Towards Self-organizing Virtual Macro Sensors

Nicola Bicocchi, Marco Mamei, Franco Zambonelli

DISMI – Università di Modena e Reggio Emilia – Via Amendola 2 – Reggio Emilia – ITALY
{nicola.bicocchi, marco.mamei, franco.zambonelli}@unimore.it

Abstract

The future mass deployment of pervasive and dense sensor network infrastructures calls for proper mechanisms to enable extracting general-purpose data from them at limited costs and in a compact way. The approach presented in this paper relies on a simple algorithm to let a sensor network self-organize a virtual partitioning in correspondence of spatial regions characterized by similar sensed patterns, and to let distributed aggregation of sensorial data take place on a per-region basis. This makes it possible to perceive the network as if it were composed of a limited number of virtual macro sensors, a feature which promises to be very suitable for a number of incoming usage scenarios.

1. Introduction

In the next few years, we will assist to an increasing deployment of sensor network systems. Rather than being closed special-purpose systems devoted to monitor specific phenomena, as they are today, they will form the basis of truly pervasive and dense shared infrastructures, publicly available for general-purpose sensing activities.

This novel perspective of usage introduces peculiar challenging requirements [MulA06, Cur05, Lu05, Cas07]. First, it is expected that the sensor network, despite being intensively used in unpredictable ways, will be able to control its energy consumption. Second, it is expected that the network will be able to provide users with expressive and compact information related to the phenomena under sensing rather than raw individual sensor data. In the presence of dense sensor networks generating huge amounts of data, dealing with the transfer and the ex-post analysis of individual sensors data can be become simply unmanageable. Third, the network should quickly answer users' requests. Since users can be highly mobile a late answer to a query can either fail to reach the user or reach him at a location where the answered information could be useless.

To tackle the dynamism and scale of such a new scenario, the idea underlying our proposal is that of delegating to the sensor network the execution of distributed self-organizing algorithms [JelMB05] that – by continuously running in the network with bounded energy costs – can enforce:

- self-partitioning of the network into spatial regions characterized by similar patterns for sensed data, via the self-organization of an overlay network.
- distributed aggregation of sensorial data on a per-region basis.

The result of this process is that a sensor network can be perceived by users as made up virtual of macro sensors, each associated to a well-characterized region of the physical environment (i.e., a region exhibiting a uniform pattern for some specific property such as a light, temperature, etc.). Within each region, each physical sensor has the local availability of aggregated data and can act as an access point to such data.

The approach based on virtual macro sensors makes it possible for multiple and mobile users to promptly access global information about the surrounding environment by simply querying the closest sensor. Also, it makes possible to effectively transfer aggregated data towards a centralized collection point in a more compact and efficient way, yet avoiding loss of information typical of global aggregation algorithms. Moreover this process is independent of the actual density, topology, and dynamics, of the underlying physical sensor network.

2. Virtual Macro Sensors

The virtual macro sensors approach considers: *(i)* a self-organized region formation algorithm to define the boundaries of each macro sensor; *(ii)* localized aggregation algorithms to provide macro sensors with regional sensorial capabilities; *(iii)* solutions to self-adapt to transitory and dynamic situations.

2.1. Region Formation

We consider a sensor network deployed in an environment in which the value v of some specific environmental property can be locally sensed. The value v could represent a temperature, a light level, or whatever

property a sensor is able to measure about its portion of the environment (see Figures 2-a).
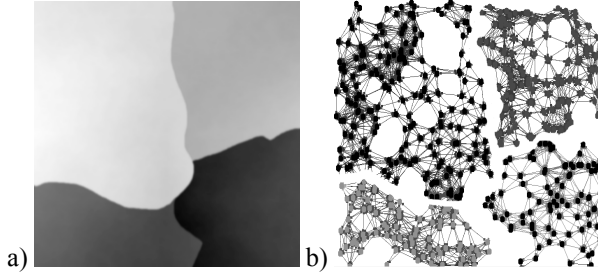

a)   b)

Figure 2. a) a scalar field with 4 regions with different values of a property *v*; b) overlay region organization leading to a partitioning into 4 small regions.

The proposed region formation algorithm has the goal of letting sensors self-organize into disjoint sets of spatial regions each characterized by "similar" measures of the property v (see Figures 2-b). Organization in regions occurs via a process of building an overlay of weighted links between neighbor nodes, such that nodes belonging to the same region have strong links, while neighbor nodes belonging to different regions have weak links. In general, the region organization can reflect some actual property of the physical space and can lead to a "logical" organization of sensors.

More in detail, let s_i and s_j be two neighbor sensors, i.e., two sensors whose distance is smaller than their wireless radio range r. Let $v(s_i)$ and $v(s_j)$ be the values of a property sensed by s_i and s_j, respectively. Let us assume that a distance function D can be defined for couples of v values. Region formation is then based on iteratively computing the value of a logical link $l(s_i,s_j)$ for each and every node of the system (see Fig. 3).

Update_link:
    if (D(v(s_i), v(s_j)) < T)
        l(s_i,s_j) = min(l(s_i,s_j) + delta, 1)
    else
        l(s_i,s_j) = max(l(s_i,s_j) - delta, 0)

Figure 3. Pseudo code for the Update_link procedure

Where: *T* is a threshold determining whether the measured values are close enough for $l(s_i,s_j)$ to be re-enforced or, otherwise, weakened; and *delta* is a value affecting the reactivity of the algorithm in updating link.

Based on the above algorithm, it is rather clear that if $D(v(s_i), v(s_j))$ is lower than the threshold T, $l(s_i,s_j)$ will rapidly converge to *1*. Otherwise it will converge to *0*. In the simplest case, one could consider two nodes s_i and s_j to be in the same region when $l(s_i,s_j)$ is over a threshold T. Transitively, two nodes s_h and s_k are defined in the

same region if and only if there is a chain of nodes such that each pair of neighbors in the chain are in the same region. For the actual execution of the algorithm, each node stores a vector describing, for each of the neighbors, the current value of the link towards it and a flag signaling the status of the link (connected or not).

The distributed execution of the algorithm is based on a sort of diffusive gossip scheme which act as a sort of continuous background activity in the sensor network: each node periodically wakes up, randomly selects a specific number or a specific percentage of its neighbors (in the case of broadcast-based wireless channels, this implies inviting a limited number of neighbors in participating in the protocol), exchanges with them the needed data, and then executes the "*Update_link*" procedure for each selected neighbors (see Fig. 4).

Do_forerever:
    Wait(t);
    Foreach(neigh[]=Select_neighbor(num_neigh))
        Data = Exchange_data();
        Update_link(data);
        // aggregation part
        Global_aggregation(); // if needed
        If(connected) Local_aggregation();
Done

Figure 4. Main algorithm. The aggregation part will be described in Sect. 2.2

From the above description it is clear that our algorithm tends to impose a uniform load to the system. Each node executes the same amount of operations. The interval *t* determines the frequency of such operations and the number of neighbors *num_neigh* selected at each round determines the communication cost of these operations. Shorter *t* or higher *num_neigh* tend to speed up the convergence of the algorithm, but increase the energy consumed by sensor per time unit (as quantified in the performance evaluation section). Therefore, one can select the appropriate values for such parameters on the basis of the application requirements.

Concerning *T*, an apparently challenging issue in our approach consists in tackling the difference between the strictly local nature of "*Update_link*" interactions and the inherently global meaning of the threshold *T*. How can two nodes evaluate the right threshold if they don't know anything about the rest of the network? Fortunately, in the vast majority of the cases, a domain expert can provide suitable and relevant thresholds to highlight the phenomena of interest and to drive the self-partitioning accordingly. For example, a difference of 5°C can be considered of relevance for a biologist to distinguish different types of landscape, and (s)he could rely on a

region-partitioning based on such a threshold. Alternatively, fire guards may be interested in much higher thresholds (e.g., 40°C) to detect anomalies. In any case, it is worth emphasizing that our approach does not prescribe the existence of a single region partitioning: multiple partitions (i.e., overlays) could be computed across the network by considering different thresholds.

In the absence of any a priori known domain data, it is still possible to define T by exploiting dynamically collected global values of the property v. For instance, in some experiments, we defined T as a portion of the whole range of values seen over the network. Using scalar values, we defined T as:

$$T = (globalMax - globalMin) * p$$

where p is a real number between 0 an 1. In this way, one can parameterize the sensibility of the algorithm by using a relative value p rather than some absolute value requiring a priori knowledge on the range of v values.

To locally acquire the $globalMax$ and $globalMin$ value at each node, one can execute a global diffusive aggregation algorithm over the whole network. The above aggregation algorithms requires minimal additional effort by nodes. In fact, one can piggyback such $globalMin$ and $globalMax$ data into the messages already exchanged by nodes.

2.2. Per-Region Aggregation

The local availability of aggregated information over the whole sensor network may be of some use independently of regions. However, globally aggregated values give very little details on the status of the network, are prone to obsolescence and high losses, and are definitely of little use for users wishing to acquire information about nearby environmental properties. For this reason, our approach mostly relies on per-region aggregation algorithms. Local aggregation algorithms enable each sensor in a region to act as a sort of access point for aggregated data in that region, and thus realize the concept of virtual macro sensor: from the application viewpoint, one can perceive a region as including a single sensor with sensorial capabilities extended to the whole region.

When regions are already formed (transitory situations will be discussed later on), computing aggregation function in a region reduces to executing a diffusive aggregation algorithm only between those couples of neighbor nodes that are in the same region (i.e., for which the l is over the T threshold). Again, computing per-region aggregation function does not introduce significant additional burden to the network.

The exchange of data between nodes can occur by piggybacking over existing messages, and the computation of local aggregation algorithms reduces to add a simple "*Local_aggregation*" function in the main body of our basic scheme (see Fig. 4).

The "*Local_aggregation*" function can include the identification of the local minimum and the local maximum of any sensed value w (other than the v property on which regions are based) within the region (computed as in the global case), as well as the calculus of the average Avg of any value w.

In our scheme, we also decided to enforce two additional peculiar aggregation functions that are of great use for facilitating the gathering of information by users.

A first aggregation function considers that each node at the border of a region (i.e., each node which has at least one virtual link l below the threshold) propagates within the region a "hop counter" initialized at 0. By having such counter re-propagated by each node on per-minimum basis, the results is that each node in a region eventually becomes aware of its distance form the closest border. We also plan to experience more sophisticated aggregation function to enable nodes to locally reach a higher-level understanding of the shape and topology of the local region, possibly relying on existing work of distributed topology recognition. What is important to note is that these kinds of topological measures are important to asses, within regions, the sensing coverage of the macro sensors.

A second aggregation function exploits a sort of per-region minimum identification towards the election of a region leader. By having each sensors exchange its unique ID with its neighbor, the minimum ID eventually recognized by each node will define the leader (and the leader itself will recognize itself as that). This is very important to give a recognizable unique identity to each macro sensor.

2.3. Transitory and Dynamic Situations

In this section we analyze the dynamic behavior of the algorithms during region formation and region re-shaping.

In general, the initial values of the virtual links l between nodes are irrelevant for region formation. Therefore, let us assume an initial situation in which all nodes are disconnected from each other (i.e., each node is a region in itself). As the algorithm will start running, nodes with similar values of v will start connecting with each other, and sets of regions with growing dimensions will start forming and possibly merge with each other, until a stable situation will be reached.

Concurrently with the above region formation

process, the local aggregation procedure starts executing as soon as two nodes gets virtually connected in the same region, and it proceeds gradually involving more and more sensors, eventually converging when a stable region situation is reached. It can be shown that the proposed aggregation algorithms do not experience problems if executed on a growing number of nodes, as in the region formation transitory. This also applies for the identification of the region leader (when two regions merge, one of the two leaders will be eventually overtaken by the other one). Similar considerations apply to the case in which new sensors are dynamically added in the system.

The case in which some existing regions shrink, (either because a confining region has expanded or because some sensor nodes have died) is a bit more complex to handle. In fact, two problems may arise: *(i)* the values computed by the local aggregation functions may no longer be valid (e.g., the former maximum may have left the region and/or the average may have changed) and the cumulative nature of aggregation does not enable them to be properly updated; *(ii)* the region leader may have exited the region.

To tackle the above two problems we are either studying and evaluating epoch based and evaporative approaches [BicMZ07].

3. Conclusions and Future Works

The proposed virtual macro sensor approach makes a sensor network self-organize into regions characterized by similar sensing patterns, so as to promote aggregation of data on a per-region basis, as if each region were monitored by a single macro sensor.

The most direct way of exploiting the virtual macro sensor approach is for supporting queries by multiple and mobile users. A user that wants to retrieve information about the surrounding will typically access the nearest sensor and query it about some local patterns of sensed data. For example, "give me the maximum temperature within 500 meters" or, by referring to some more logical environmental concept, "give me the average temperature in this room". At this point, in most of the cases, the queried sensor can immediately answer to the user without further burdening the network, independently of the number of mobile users.

To test the effectiveness of the approach, we have experimented it both in a simulation environment and in a small sensor network test bed.

Simulations have been built over the Repast simulation framework [http://repast.sourceforge.net/]. We have conducted several experiments with sensor networks of different sizes and densities, immersed in different types of scalar fields, always obtaining similar qualitative and quantitative results. The proposed approach seems to be effective and scalable and in certain situations could represents a concrete alternative to routing trees based approaches.

Despite the encouraging results obtained so far, we are aware of a number of limitations of our work, subjects of our current research work. These include: generalizing the approach to support multiple overlays and general-purpose queries; exploring inter-region algorithms to support more global queries; defining algorithms to promote the building of high-level knowledge about the global structure and properties of the of virtual macro sensors network.

References

[BicMZ07] N. Biccocchi, M. Mamei, F.Zambonelli., "Self-organizing Virtual Macro Sensors", Techical Report DISMI-UNIMORE, Feb 2007.

[Cas07] G. Castelli, A. Rosi, M. Mamei, F. Zambonelli, "A Simple Model and Infrastructure for Context-aware Browsing of the World", International Conference on Pervasive Computing and Communication, New York (NY), 2007.

[Cur05] C. Curino, M. Giani, M. Giorgetta, A. Giusti, A. Murphy, G. Picco, "Mobile Data Collection in Sensor Networks: The TinyLime Middleware", Journal of Pervasive and Mobile Computing, (4)1:446-469, 2005.

[JelMB05] M. Jelasity, A. Montresor, O. Babaoglu, "Gossip-based Aggregation in Large Dynamic Networks", ACM Transactions on Computer Systems, 23(3):219-252, 2005.

[Lu05] C. Lu, G. Xing, O. Chipara, C. Fok, S. Bhattacharya, "A Spatiotemporal Query Service for Mobile Users in Sensor Networks", International Conference on Distributed Computing Systems, Columbus (OH), 2005.

[MulA06] R. Müller, G. Alonso, "Shared Queries in Sensor Networks for Multi-User Support", Technical Report 508, ETH Zürich (CH), 2005.