Distributed Context-Aware Affinity Propagation Clustering in Wireless Sensor Networks

Mahmoud ElGammal
Bradley Department of Electrical and Computer Engineering, Virginia Tech
gammal@vt.edu

Mohamed ElToweissy[1]
Cyber Security Research
Pacific Northwest National Laboratory
mohamed.eltoweissy@pnl.gov

***Abstract*—We foresee the need for dynamically clustering nodes in Wireless Sensor Networks (WSNs) according to a multitude of disparate co-existing contexts. To this end, we propose a distributed, low-overhead context-aware clustering protocol for WSNs. We employ Affinity Propagation (AP) for clustering nodes based on multiple criteria including location, residual energy, and contextual data sensed from the environment. We propose a novel approach for context representation based on potential fields. We discuss the integration of our context representation model with AP and demonstrate using simulation the effectiveness and proficiency of the proposed protocol in satisfying its intended objectives.**

***Index Terms*—affinity propagation, context awareness, distributed clustering, potential fields, wireless sensor networks.**

I. INTRODUCTION

WIRELESS Sensor Networks (WSN) are getting more popular everyday, and they are being deployed to serve an ever increasing number of applications. A single WSN can be responsible for monitoring events belonging to multiple contexts concurrently, and serve users with different roles. Clustering is necessary in WSNs for improving scalability and for efficient use of resources. However, most of the current clustering algorithms for WSNs focus on a single aspect such as data aggregation (as in [1] and [2]), conserving energy, or security against threats [3]. A clustering algorithm equipped with a means for representing dynamic contextual data is needed.

The main contributions we present in this paper are: (1) a low-overhead, distributed, context-aware clustering protocol built around Affinity Propagation (AP) [4], and (2) a novel approach for representing spatiotemporal contextual events using *potential fields*. AP is a relatively new clustering technique that has been shown to possess several advantages over long-standing algorithms such as K-means, particularly in terms of quality of clustering and multi-criteria support [4]. The potential field representation allows us to quantify the effect perceived by sensor nodes of nearby events, a crucial ingredient for successful integration with the clustering algorithm in order to achieve context awareness.

The objectives of the presented protocol is to achieve high quality clustering with low overhead, extending the network lifetime, and having the ability to accommodate concurrent events pertaining to different contexts.

For a WSN operating in a complex environment where many contexts have to be catered for simultaneously, a flexible clustering algorithm capable of accommodating all such contexts is needed. AP possesses such capability, but its quadratic complexity and the fact that it is centralized by nature represent an obstacle to its employment in a WSN. Zhang et al [5] propose a modification to AP that removes both restrictions. The proposed algorithm (Hi-WAP) works in a hierarchical manner by first dividing the data points (or sensors in our case) into subsets. AP is then executed on each subset individually, and the exemplars (cluster heads) of each subset are identified. Finally, only those points identified as exemplars contribute to the final *weighted* AP run. This results in a great reduction in execution time with a minimal toll on clustering quality. In our previous work [6], we studied the feasibility of Hi-WAP for clustering of wireless sensor nodes, and presented a location-aware adaptation for WSNs which we coined *Location-aware Affinity Propagation* (LAP). We extend our previous work here by presenting a concrete realization of an AP-based clustering protocol.

As a possible application for our work, we consider a hypothetical border security system to detect and prevent illegal border crossings, drug trafficking, smuggling, and other similar criminal activities. The application relies on WSN comprised of a large array of sensors such as electro optical cameras, infrared cameras, vibration sensors and explosive-material sensors. The initial mission of the WSN is to observe the deployment area and report any suspicious activities.

For a more concrete scenario, consider that an infrared-based vehicle recognition algorithm in the WSN detects that a human trafficking attempt is taking place using a minivan. The WSN alerts an agent with the Border Patrol Tactical Team, via his PDA. After an automated evaluation of a pursuit-and-capture mission, a few members of the Special Response Team are assigned to the mission and instructed to use a High Mobility Multipurpose Wheeled Vehicle (HMMWV). The HMMWV is equipped with high performance computers and

This work was supported in part by a National Science Foundation NeTS award 0721523.

[1] This work was done in part while ElToweissy was affiliated with the Bradley Department of Electrical and Computer Engineering at Virginia Tech.

COLLABORATECOM 2010, October 9-12, Chicago, USA

DOI 10.4108/icst.collaboratecom.2010.54

high power communication devices. Moreover, a number of Unmanned Ground Vehicles (UGVs) are deployed to facilitate the mission. Meanwhile, a number of Unmanned Aerial Vehicles (UAVs) are detected while penetrating a section of the border, in one of many attempts by the perpetrators to find a vulnerable spot. The WSN tracks the UAVs and predicts their future locations, and guides a number of Counter-UAV Vehicles (CUAVV) to suitable locations where they can shoot down the UAVs. Finally, because the WSN indicates the possible existence of explosives and weapons, the Border Patrol Search, Trauma, and Rescue Team is contacted to ensure that an ambulance helicopter and a fire truck are called to be on site. As the team begins the pursuit-and-capture mission, under the constant attack on the WSN by the intruders to inhibit the network functionality, the WSN autonomously morphs its clustering in order to: (1) seamlessly assimilate the HMMWV, CUAVVs, the UGVs and the helicopter as new nodes with higher capabilities; (2) cope with the context of the new mission; (3) minimize the risk of being damaged by the intruders, and (4) In situations where the network has to deal with multiple concurrent events, the available resources are assessed and then assigned to handle them, or in case of resource deficiency, the events are prioritized and the network calls for more external resources.

The remainder of this paper is organized as follows. In Section II we present the protocol in detail. In Section III, we discuss how the protocol objectives were achieved and provide performance figures. Finally, in Section IV, we state our conclusions and discuss future work.

II. Context-aware Affinity Propagation Clustering Protocol

A. Overview

The network deployment area in LAP is perceived as a square grid with k x k cells [6]. The algorithm has to be executed over a number of steps or *phases* depending on the depth of the hierarchy, where the area of each grid cell is squared in each new phase. A coordinator node (*S)* is responsible for initiating and concluding each phase. During each phase, nodes within each cell exchange messages with each other using geocasting. Then, an execution node E_i is selected in each cell to run AP over the data received from all other nodes in the cell. The results are then sent back to *S*, which initiates the next phase upon receiving the results from all execution nodes. Only nodes selected as cluster heads in a phase contribute in the next phase, and the algorithm terminates when the grid contains only one cell. Fig. 1 shows an overview of the process.

B. Protocol Description

In this section we provide a detailed description of the LAP protocol, where we discuss the different roles assumed by the nodes, the messages exchanged between them, and the actions taken by them upon receiving a message. Table 1 is a legend of the notation that will be used henceforth.

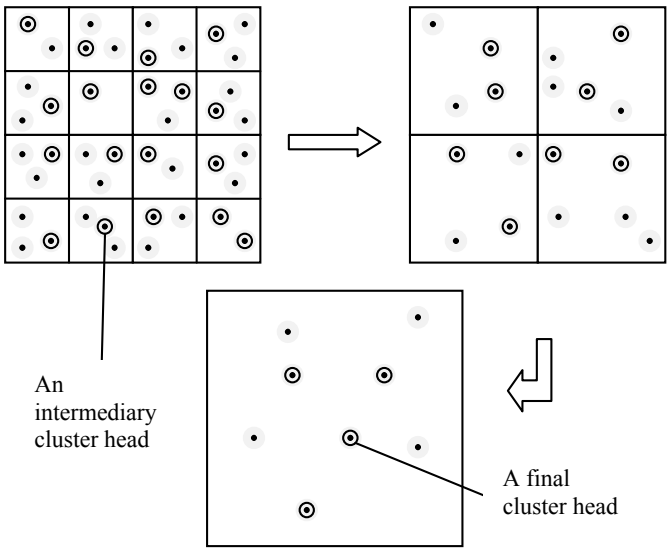

Fig. 1. A sample run of LAP. Grid cell area is squared after each phase, and only nodes selected as exemplars in one phase participate in the next one.

Symbol	Meaning
k	Initial number of cells in each dimension in the grid. k is divided by two in each new phase until it is equal to 1. It also represents the hierarchy depth in the Hi-WAP algorithm.
p	Current phase number $\in\{1,..,1+\log_2 k\}$
k_p^2	Total number of cells in the grid during phase p.
S	The coordinator node.
$E_p(i)$	Execution node in cell i during phase p. $i \in \{1,..,k_p^2\}$
$n_p(i)$	The set of nodes selected as cluster heads in grid cell i in the phase $p-1$. Initially, this includes all nodes in the grid cell.

Table 1. Notation

For the protocol to be executed correctly, we make the following assumptions about the underlying network:

- Nodes are equipped with GPS hardware, or are otherwise aware of their locations, allowing them to determine the grid cell they lie within.
- Each node knows its neighbors in the same grid cell. This can be achieved using any neighbor discovery protocol, such as [7], [8] or [9]. It is sufficient to run the protocol once after the sensors are deployed. Only neighbors within the initial cells need to be identified (i.e. during the first phase).
- Like in [10] and [11], we assume that each cluster head broadcasts a TDMA schedule for its nodes to follow, and that Direct Sequence Spread Spectrum (DSSS) is used to mitigate collisions.

The protocol consists of the following steps, which are all overseen by *S*. The initial selection of *S* is arbitrary.

I. **Phase Initialization Step**: The protocol starts by *S*

broadcasting a `PhaseExecute` message that contains the phase number, p, and the size of the grid cell that should be used in the phase.

II. **AP Matrix Acquisition Step**: upon receiving this message, each node in $n_p(i)$ collects the application-specific parameters needed for clustering from all other nodes in the cell. This is achieved by having each node geocast these parameters to all other nodes in its cell encapsulated in an `APValues` message. Each node then waits until it has collected similar messages from all neighbors in its cell. If a neighbor fails to send a message within a certain timeout, possibly due to energy depletion, it is considered dead and isn't waited for in any subsequent runs of the protocol. The exact parameters included in the message vary according to the application, and will be discussed in section IV.

III. **AP Execution Step**: The `APValues` message includes application-specific data relevant to clustering, in addition to information about the residual energy in the source node. When all the nodes in $n_p(i)$ have responded (or after the timeout has been reached), this information allows the node with the highest residual energy to identify itself, as it is responsible for executing the Weighted-AP algorithm on the collected data, which results in identifying cluster heads within the cell. The result is sent to node S as a `PhaseDone` message.

IV. **Phase Conclusion Step**: Once S receives the `PhaseDone` message from all execution nodes (or after a timeout proportional to the number of nodes), the current phase is considered complete. If k_p^2 is equal to 1, the algorithm is complete and the protocol proceeds to the next and final step, otherwise the grid cell size is doubled, p is incremented, and we start again from step I. Only nodes selected as cluster heads in the last phase contribute in the new one. The other nodes can simply ignore any messages until the global cluster heads are identified in the final step.

V. **Result Broadcasting Step**: By now, S knows the global cluster heads, so it composes and broadcasts a `Results` message that identifies the cluster head of each node in the network. Once received by a node, it can immediately determine its role in the final clustering.

C. Practical Considerations

In order for the protocol to operate efficiently and deterministically, there are certain practical considerations that have to be made. For instance, the initial grid cell size is directly proportional to power utilization, while it's inversely proportional to the number of phases. On the one hand, since nodes have to geocast messages within their grid cell, the cell size has to be small enough such that it allows each node to reach its same-cell neighbors without using too much power. On the other hand, the cell size shouldn't be too small that it results in a high number of phases, which affects the clustering quality negatively as we have shown in [6]. In the following section we show that by using a suitable value for k, the geocasting overhead is mitigated, allowing LAP to outperform two other leading clustering protocols.

Another important consideration is reliability. The coordinator node S in particular represents a single point of failure in the protocol. This, however, can be solved using several techniques, and it's up to the specific application to choose which is more suitable:

- S could be a special node with abundant energy and immunity to physical attacks.
- It could be the base station itself, if it's appropriately located relative to the other nodes.
- S could be a group of nodes, with only one of them initially active, and the others being passive ranked replicas. Nodes contributing in the protocol would use multicasting to send data to S. All the replicas would store the data, but only the active node sends out LAP messages. All the nodes in the group periodically send heartbeat messages, and in case the active node fails, the next available node by rank is activated.

III. Protocol Objectives

Attempting to tackle the problem of clustering in wireless sensor networks poses additional challenges that don't manifest themselves in offline clustering. Nodes have to exchange messages between themselves and execute the clustering algorithm on the collected data. This could represent a major energy drain, adding more strain to the already limited power resources of wireless sensor networks.

Moreover, the clustering protocol should utilize node resources as evenly as possible. Over utilization of resources on a subset of nodes results in a skew in node lifetime, shortening the longevity of the entire network and, depending on the application, could severely affect the network's capacity to function correctly. For example, in an environment monitoring application, the overuse of nodes in a certain area in the deployment field could result in the lack of information about that particular area much earlier than other parts of the field.

A paramount objective that we aim to achieve as well is *generality*. The solution we propose here is geared towards multi-context applications where the network has to fulfill different – even conflicting – goals simultaneously. We achieve that by representing contextual data (events of interest to the sensor network) as *potential fields*, and interweaving them with the Affinity Propagation algorithm. This section discusses each of these objectives in detail.

A. Protocol Overhead

We have made use of certain properties of the AP clustering algorithm as well as of the underlying network in order to minimize the protocol overhead as much as possible.

A key property of LAP (and also of Hi-WAP) is the fact that the number of nodes contributing to each phase decrease exponentially as the algorithm progresses. In Weighted Affinity Propagation, nodes in a lower level in the hierarchy are represented by their exemplars in the upper level. This results in great reductions as the algorithm proceeds from one phase to the next, which translate into energy savings as the number of exchanged messages is also reduced. Almost no energy is exerted at all by any nodes except the exemplars and the nodes needed to route messages between them. Fig. 2 shows the reduction in the number of contributing nodes as the phase number increases.

The total number of messages generated by the protocol is another important overhead metric. Fig. 3 shows the ratio of the total number of messages to the total number of nodes. The ratio seems to follow a logarithmic curve, where the overhead relatively decreases as the number of nodes increase.

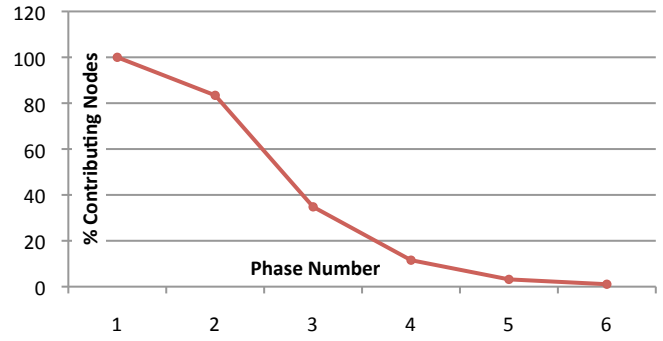

Fig. 2. A sample run with 1200 nodes and k=5. With the exception of phase 1, the percentage of nodes contributing to each phase is reduced by more than 50% in each subsequent phase.

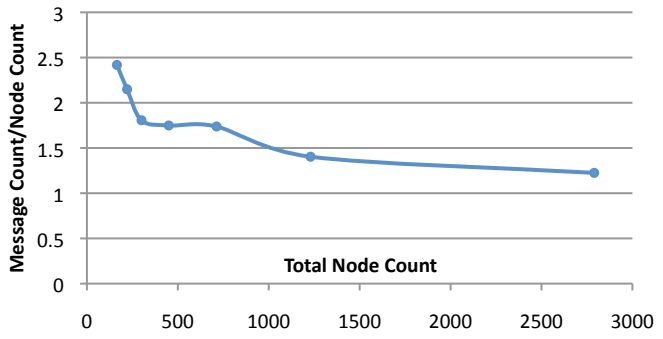

Fig. 3. The total number of messages is inversely proportional to the total number of nodes. The curve is logarithmic with an asymptote of 1. These figures were obtained for k=4 and a deployment area of 1000x1000.

Another factor that helps to limit the number of messages and aids in saving node energy is geocasting. By restricting the exchange of `APValues` messages to a single grid cell, nodes can use lower power levels to transmit messages within their own cells, resulting in better power efficiency and less network congestion.

B. Network Longevity

Optimal network lifetime is the result of two practices: the use of efficient algorithms, and unbiased utilization of node resources. The former has already been discussed in *A*. The latter, however, is achieved in our scheme by rotating the different roles in the protocol over the nodes according to their residual energy levels. The most power-demanding role in the protocol is that of execution nodes.

In Step III of the protocol, nodes in each cell exchange `APValues` messages, which include information about the residual energy on the source node. Once a node receives all `APValues` messages from its cell neighbors, it can determine whether it is the one with the highest level of residual energy, and if it is, it immediately executes AP on the data it has. The next time the protocol is started, another node will probably be a better candidate to execute the algorithm, and as the process is repeated multiple times the variance between the residual energy levels is always kept at a reasonable level. The selection of the coordinator node *S* is also done similarly.

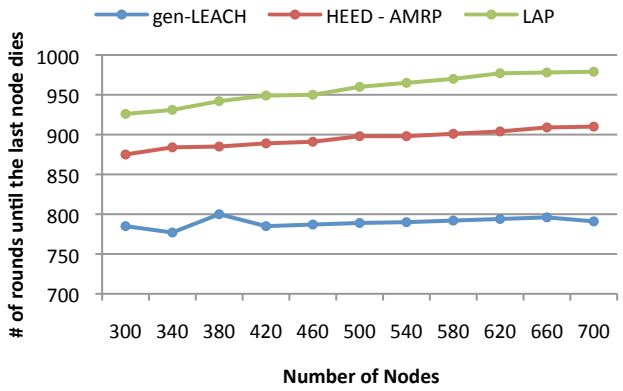

Fig. 4. Network lifetime under different clustering protocols for different deployment sizes *(k = 8)*.

Fig. 4 shows a comparison between the network lifetime achievable using LAP and two other well-known clustering protocols: *generalized LEACH* [12], and *Average Minimum Reachability Power (AMRP) HEED* [10]. The simulation parameters are identical to those in [10] and are repeated in Table 2 for convenience. The sensors are deployed in a 100x100 m^2 area with a base station located at (50,175). During each TDM frame, each node collects data and sends a data packet to its cluster head, and each cluster head fuses those packets into one message and sends it to the base station. Clustering is triggered each round, where a round is 5 TDM frames. The results show that LAP consistently achieves a longer network lifetime than both LEACH and HEED.

Type	Parameter	Value
Network	Network grid	From (0,0) to (100,100)
	Sink	At (50,175)
	Initial Energy	2 J/Node
Application	Data packet size	100 bytes
	Packet header size	25 bytes
	Round	5 TDM frames

Radio Model	E_{elec}	50 *nJ/bit*
	ε_{fs}	10 $pJ/bit/m^2$
	ε_{mp}	0.0013 $pJ/bit/m^4$
	E_{fusion}	5 *nJ/bit/signal*
	Threshold distance d_0	75 m

Table 2. Simulation Parameters. Please refer to [10] for a detailed description of each parameter.

C. Generality

The primary target we consider for the utilization of this protocol are highly dynamic pervasive cyberspaces; physical environments with embedded information appliances that are gracefully integrated into and coordinated with the physical resources to better serve human users, and where events are of a dynamic spatio-temporal nature. We focus our attention on scenarios involving multiple concurrent events pertaining to different contexts. In such situations clustering must be based on multiple criteria, and conflict resolution has to be performed when concurrent tasks tax network resources in a conflicting manner. What is needed is a context representation model that satisfies the following criteria:

- It should provide a means of quantifying network events.
- It should simplify the process of conflict resolution between events belonging to different contexts.
- It should be easy to integrate with the Affinity Propagation algorithm.

Given the resource scarcity that characterizes most sensor networks, the network must utilize each individual node in the most efficient manner while performing its function as optimally as possible. This means that multiple events have to be dealt with differently according to their location, urgency, and possibly other attributes. It is important to choose a representation that allows individual nodes to efficiently choose the events to cover, form clusters with each other, and guarantee good coverage for the entire network. The potential field representation has several merits that would help us achieve these goals as we are going to show shortly.

The term *potential field* refers to a force associated with each point of space-time. In physics, such fields exist as a result of the presence of certain objects, such as a magnet or an electron, and the strength of the force exerted on objects residing in the field is a function of the distance between the actor and the acted-upon objects.

Potential fields have been used extensively in many areas, but particularly in electromagnetics and robotics. In wireless sensor networks, [13] shows how mobile sensor nodes can be represented as having similarly charged potential fields which guide their movement in order to achieve better coverage. In our work, however, our goal is to model events (contextual information) as potential fields acting on sensor nodes and affecting many aspects of their operation such as clustering, data aggregation, and so forth.

Quantifying Events

We perceive an event as an incident occurring at a certain location and which can only be detected by nodes within a certain range. Typically, the ability of a node to detect an event weakens as the distance separating them increases. This can be accurately represented as a potential field with a force that reaches its peak at the center of the area covered by the event and which decays until it becomes undetectable as we move away as shown in Fig. 5. The gradient of the force can be represented mathematically by a multitude of functions. We have chosen a parabolic curve that is a function of the square of the distance to the event location. The task of actually detecting the events, identifying their types, and tracking their locations is outside the scope of this paper, but [14] and [15] can be consulted for possible implementations of such functionality.

Conflict Resolution

In our proposed scheme, we would like each node to behave as independently as possible in order to minimize the communication overhead, but without compromising the functionality of the network as a whole. In other words, we would like to guide nodes to take local decisions that collectively serve global goals. This can be achieved by assigning different profiles to sensor nodes. Such profiles are mutable, and they change in reaction to external events or decisions taken by neighbor nodes. This allows each node to view its surroundings from its own perspective and take independent decisions, and also adapt itself to the feedback it receives from its neighbors in order to enhance the performance of the entire system.

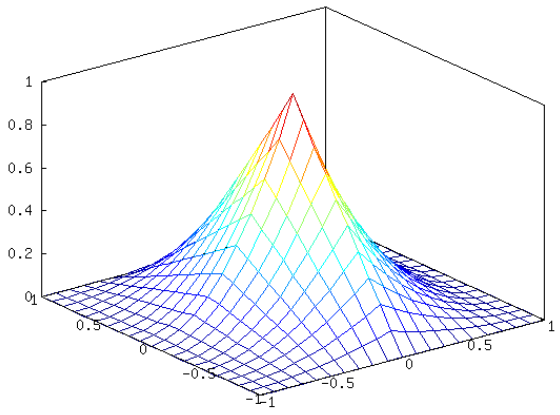

Fig 5: A normalized potential field: center at *(0,0)*, coverage extends from *(-1,-1)* to *(1,1)*, and maximum strength is 1. The field strength is calculated as $(1-d)^2$ where d is the distance between each point and (0,0).

Node profiles determine which event types a node should be involved in servicing, which should be avoided, and the type of manipulation the node should perform on the potential fields representing the detected events to arrive at its decisions. Sensor nodes, each according to its profile, must calculate the force exerted on them by each interesting event they can detect. From that point, many operations are made possible thanks to the fact that a quantitative representation of

each event is now available. For instance, a node could choose the event to report according to its strength and to the weight assigned to the event type in the node's profile, or it could add all the force vectors it can detect, and depending on the angle of the resultant, decide which cluster to join, or which direction to move into if it's a mobile node. Events in the *threat context* can be represented as having negative potential, causing mobile nodes to move away from them. It can be seen that all such operations reduce to basic vector arithmetic that can be performed efficiently even by resource-limited nodes.

Integration With Affinity Propagation

The AP algorithm basically relies on the acquisition of a similarity matrix, where the value of a cell *s(i,j)* expresses how similar node *i* is to node *j*. The similarities are computed by the execution node using the data it receives from the other nodes in the cell. The algorithm converges after a certain number of iterations, during which, two types of values are computed for each pair of nodes: the *responsibility* value *r(i,k)* represents the accumulated evidence for how well suited point *k* is to serve as an exemplar for point *i*, while the *availability* value *a(i,k)* reflects the accumulated evidence of how appropriate it would be for point *i* to choose *k* as its exemplar, given the support *k* has from other candidate cluster members. Another important piece of information that the similarity matrix holds is the self-similarity *s(i,i)*, also called *preference*, which influences the number of clusters generated by the algorithm (for a comprehensive description of the algorithm please refer to [4]).

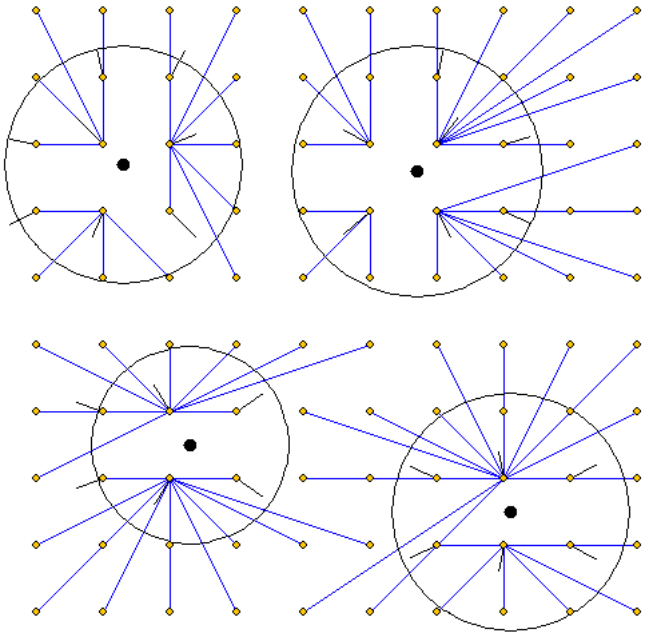

Fig. 6. Nodes laid out as a grid and clustered according to their proximity to events (represented by the circles.) Lines emanating from nodes lying within the potential fields represent the forces acting on them.

While nodes cannot control the responsibility and availability values directly, they could influence the similarities through the values they encapsulate in `APValues` messages, and also set the preferences directly. Preferences are derived from the potential field strengths, and normalization is applied to make the values in the similarity matrix homogeneous. Usually, when the number of clusters is not known, preferences are set to the mean value of all similarities. By offsetting that value by the potential field strength, we effectively increase the chances of those nodes lying in the vicinity of events to become cluster heads. Equation 1 shows how the preference value is calculated. N is the total number of nodes, and *force(i)* is the vector sum of forces acting on node *i*. Fig. 6 shows the resulting clusters in a sample run of the protocol.

$$\mu_{sim} = \sum_{i=1}^{N} \sum_{j=1, j\neq i}^{N} \frac{similarity(i,j)}{N^2 - N} \quad (1)$$

$$\mu_{force} = \sum_{i=1}^{N} \frac{force(i)}{N} \quad (2)$$

$$similarity(i,i) = \mu_{sim} - force(i)\frac{\mu_{sim}}{\mu_{force}} \quad (3)$$

IV. Conclusion and Future Work

Context aware clustering allows us to group nodes in a WSN based on all of the information available to the network and not just according to predefined static parameters such as location or energy. We have made use of the modifications proposed for AP in [5] that address the issues of centrality and high time complexity and adapted then extended the algorithm for WSNs. We presented a distributed protocol that coordinates the execution of the algorithm on the nodes, and incorporated context awareness into the algorithm by employing a novel context modeling tool enabling us to quantify the effect of events on sensor nodes. Our work is ongoing on testing this solution in various real-life application scenarios. Among the issues we are exploring is the definition of node profiles, where each node would have different sensitivities to different event types according to their capabilities. Such profiles could be dynamic, where nodes change their behavior according to the state of the network. Defining node behavior is another interesting area that we are exploring. One of the solutions we are studying is using fuzzy logic to determine the action a node should take. The representation of events as potential fields would be very convenient, as field strengths can be used as direct inputs to the fuzzy logic functions defining node behavior.

References

1. Huifang, C., H. Mineno, and T. Mizuno. *A Meta-Data-Based Data Aggregation Scheme in Clustering Wireless Sensor Networks*. in *Mobile Data Management, 2006. MDM 2006. 7th International Conference on*. 2006.
2. Yoon, S. and C. Shahabi, *The Clustered AGgregation (CAG) technique leveraging spatial and temporal correlations in wireless sensor networks.* ACM Trans. Sen. Netw., 2007. **3**(1): p. 3.

3. Blace, R., M. Eltoweissy, and W. Abd-Almageed, *Threat-Aware Clustering in Wireless Sensor Networks*. 2008. p. 1-12.
4. Frey, B.J. and D. Dueck, *Clustering by Passing Messages Between Data Points.* 2007: p. 1-23.
5. Zhang, X., C. Furtlehner, and M. Sebag, *Distributed and Incremental Clustering Based on Weighted Affinity Propagation.* 2008: p. 1-12.
6. ElGammal, M. and M. Eltoweissy. *Location-Aware Affinity Propagation Clustering in Wireless Sensor Networks*. in *Wireless and Mobile Computing, Networking and Communications, 2009. WIMOB 2009. IEEE International Conference on*. 2009.
7. Borbash, S.A., A. Ephremides, and M.J. McGlynn, *An asynchronous neighbor discovery algorithm for wireless sensor networks.* Ad Hoc Networks, 2007.
8. Kohvakka, M., et al., *Energy-efficient neighbor discovery protocol for mobile wireless sensor networks.* Ad Hoc Networks, 2009.
9. Hamida, E.B., G. Chelius, and E. Fleury, *Revisiting neighbor discovery with interferences consideration.* 2006.
10. Younis, O. and S. Fahmy, *HEED - A Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad Hoc Sensor Networks.* 2009: p. 1-14.
11. Heinzelman, W.R. and H. Balakrishnan, *Energy-Efficient Communication Protocol for Wireless Microsensor Networks.* 2000(0061).
12. Heinzelman, W.B., *Application-Specific Protocol Architecture for Wireless Networks.* 2007(0060): p. 1-154.
13. Howard, A., M.J. Mataric, and G.S. Sukhatme, *Mobile Sensor Network Deployment using Potential Fields: A Distributed, Scalable Solution to the Area Coverage Problem.* 2009: p. 1-10.
14. Cao, Q., et al., *Analysis of target detection performance for wireless sensor networks.* Distributed Computing in Sensor, (0084).
15. Wittenburg, G., et al., *A system for distributed event detection in wireless sensor networks.* IPSN '10: Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks, 2010(0100).