Biologically Inspired Methods for Organizing Distributed Services on Sensor Networks

Franz J. Rammig, Tales Heimfarth, and Peter Janacik
Heinz Nixdorf Institute
Universität Paderborn
Paderborn
Germany
{franz, tales, pjanacik}@upb.de

Abstract We propose to make use of a completely distributed way of implementing OS services for wireless sensor networks (WSN). I.e. instead of having an instance of the OS on each node of a WSN the services of the OS are distributed over the WSN. Of course this approach implies specific challenges. Two of them are discussed in the paper: Migration of services to nodes such that the overall communication costs are minimized and forming clusters with the tendency to concentrate service requests inside the clusters and at the same time minimizing intra-cluster communication. For both problems biologically inspired solutions are discussed. Service migration is mapped on an Ant Colony Optimization (ACO) technique while as a clustering heuristics Division of Labor in swarms of social insects is used.

Keywords: Wireless Sensor Networks, clustering, service migration, Ant Colony Algorithms

1 Introduction

Wireless ad hoc networks enable a myriad of novel applications ranging from human-embedded sensing to ocean data monitoring. Given current hardware limitations of wireless nodes, e.g. commercial off-the-shelf sensor nodes, approaches for the management of ad hoc networks have to be designed to function using only a low amount of resources and communication overhead. Future *Wireless Sensor Networks (WSN)* will require increasingly comfortable operating system services. This demand may go beyond the level as supported by state of the art approaches like *TinyOS*, *Mantis*, or *ContikiOS*. We propose to make use of a completely distributed way of implementing such OS services. I.e. instead of having an instance of the OS on each node of a WSN the services of the OS are distributed over the WSN. Our *NanoOS* [9] is a small distributed OS for sensor networks. In order to provide more functionality on hardware constrained nodes, we are distributing the OS and application services among the nodes of the network. We organize the network in clusters. After this, each

Dagstuhl Seminar Proceedings 08141
Organic Computing - Controlled Self-organization
http://drops.dagstuhl.de/opus/volltexte/2008/1565

OS and application instance (a set of services) is distributed inside one cluster. We set the resource requirement (*q*) to the worst-case resource utilization of one instance of the OS and application. Therefore, it is guaranteed that each cluster has enough resources for an instance of our distributed NanoOS. Of course this approach implies new challenges. Two of them are discussed in the presentation: The first one copes with forming clusters with the tendency to concentrate service requests inside the clusters and at the same time minimizing intra-cluster communication. The second one addresses migration of services to nodes in such a way that the overall communication costs are minimized. For both problems ant colony inspired solutions are discussed.
The total biomass of ants on earth is more or less the same as the biomass of mankind. Ants can be seen as one of the most advanced examples of social bio-systems. Ant colonies can be interpreted as a specific kind of an organism, forming an interesting compromise between simple swarms of single cell life and highly organized multi-cell systems (e.g. mammals) where most cells are fixed at a specific location and play a specific role. Differently from these two extremes in an ant colony the individual constituent (an ant) is a multi-cell object, mobile, intelligent to a certain degree, but closely embedded into a global collaborative scheme. *Ant Colony Optimization (ACO)* is a cooperative meta-heuristic being successfully applied to various combinatorial optimization problems. Ants tend to find the shortest path from their nests to a food source in a relatively short time. For doing so, they communicate in an indirect manner, called *stigmergy*. Moving ants deposit traces of pheromone on their trail. On the other hand, ants have the tendency to follow trails which are marked by pheromone. This establishes a positive feedback which makes a marked trail even more attractive. Evaporation of pheromone establishes a negative feedback. When alternative trails are chosen randomly in the beginning, the pheromone level of a path is inverse proportional to the path's length with high probability. Dorigo et al. [7] were among the first to apply ACO to graph-related optimization problems like the *Traveling Salesman Problem (TSP).*

2 Clustering

There are several clustering algorithms that aim to find the Maximum independent set (MIS) of a network modeled as an undirected graph. This is often combined with the dominance property, which means that the following properties should be satisfied: *independence* (no two clusterheads can be neighbors) and *dominance* (every ordinary node has at least a clusterhead as direct neighbor). There are several algorithms that satisfy these properties resulting in a 1-hop distance to the clusterheads ([2, 3, 5, 8, 13]). The *Max-Min D-Cluster Formation* [1] aims to construct the cluster with nodes at most *d* hops away from the clusterhead. The *Budget Approach* [14] tries to divide the ad hoc network into a set of clusters whose size is close to a predefined one. In contrast to above heuristics as well as the upper and under bound approach [4], we need a clustering algorithm that pursues a different objective: all clusters should possess a minimum amount of resources (i.e., the under bound limit is not given by a

size in nodes, but by an amount of resources). The optimization objective is to minimize the internal cluster communication cost.
The ad hoc network is modeled by an undirected graph $G = (V,E)$, where V is the set of wireless nodes and an edge $\{u, v\} \in E$ if and only if a communication link is established between nodes $u \in V$ and $v \in V$. Each node $v \in V$ has a unique identifier (ID_v).
For each link, a weighting function attributes a positive weight $w : E \rightarrow [0, 1]$ that represents the quality of a wireless link. In [11], the used method to estimate the quality of a wireless link is presented. We define for each edge not in the graph $w(u, v) = \infty$.
For each node, an additional weighting function r is responsible to characterize the amount of resources available in the node. $r: E \rightarrow \Re^*$. This models the resource capacity of the node.
We aim to create multihop clusters with a minimum amount of resources per cluster minimizing the intra-cluster communication cost. We are following an approach based on division of labor in colonies of social insects like in the *Pheidole* genus. The basic idea in this case is to treat each node of a WSN either as a "*major*" ant or a "*minor*" one. A major represents a clusterhead which means a higher workload while the minors are member nodes of clusters. The main power of the approach is originating from the built-in elasticity. Both types of species have a certain threshold to become major or minor. On the other hand they are stimulated by received signals. Whenever the strength of such signals is above a certain threshold the role of a major may change to a minor or vice versa. Typical stimuli signals are signal strengths of received messages, frequency of received messages, etc. Thresholds are established e.g. by the power reserve of a node. A cluster head with flattening power resources has a tendency to become a minor (member node), an "isolated" member node to become a cluster head; see [10] for more details.

2.1 Formulation of the Problem

The optimization problem to be solved is given as follows:
Input: A graph with weighted nodes and links (G,w,r) and a resource requirement $q \in \Re^*$ that must hold in each cluster.
Constraints: For every input instance we require that (1) the set of clusters covers V, (2) the clusters are disjoint, (3) the clusters have to be connected internally, and (4) per cluster there is a minimum amount of resources.
Goal: Minimize the sum of intra-cluster communication costs, i.e. the sum of the link costs over all clusters. In each cluster, this cost is given by the sum of the link costs from every node to all other ones using shortest paths. For a formal definition of the optimization problem see [10].

2.2 Clusterhead Selection

In the initial state, all nodes of the network are ordinary nodes, i.e., there is no cluster structure in the network. The variable $state_v$ describes the actual state of a node v ($state_v \in$ *{Clusterhead(CH), Member(Me) , Nonmember(Nm)}*) and c_i is set of the current members of the cluster i . There is a response function T which is responsible for the transition from an ordinary (*Nm*) node to clusterhead (*CH*). It has two arguments: a threshold τ and a stimulus σ. The threshold indicates how appropriate a node is for a role. Smaller τ value means that the node is very well suited to carry out the role of a clusterhead. The idea of this threshold function is that nodes with high energy level and high connectivity are good candidates for becoming elected as clusterhead. The energy is an important factor because clusterheads get assigned administrative tasks within the cluster and have a special status in the network. Good connectivity comes from the greedy assumption that starting a cluster from well-connected nodes will result in a relatively small clustering cost. The underlying idea for the stimulus values is that nodes that are not belonging to any cluster for a longer period of time and nodes without clusters in their vicinity should have a higher stimulus to become clusterhead. Some nodes will start to change the role to clusterhead based on the stimulus function. When a node decides to be clusterhead, it selects a random ClusterID. Now the membership election is initiated.

2.3 Member Selection

A couple of properties control the membership selection process: (1) The *distance to the closest node already in the cluster* (small distance helps to reduce the communication cost within the cluster), (2) the *distance to the clusterhead* (responsible for shaping the cluster in order to constrain its diameter), (3) *connectivity to nonmembers* (high connectivity helps to reduce the communication cost within the cluster when there is a lot of resources still missing in the cluster), (4) *connectivity to members of the cluster* (high connectivity helps to reduce the communication cost within the cluster, and (5) the *resource availability of the node* (nodes with higher resource availability will reduce the cost of the cluster to a greater extent since fewer of them are needed).

The *Membership-Select* algorithm is an incremental process, i.e., at beginning, the cluster has just the clusterhead (*CH*) node and during the clustering process, more and more nodes are added until the cluster achieves an appropriate size. When a node becomes part of the cluster (including the clusterhead), immediately a message is broadcasted to the neighboring nodes signalizing the new status and requesting new members (*Call Members* message). Each nonmember and deciding node d that receives this message changes its state to *deciding*. Deciding nodes are the potential new members of the cluster. Nevertheless, not all nodes are the best choice to be included into the cluster. In order to privilege nodes potentially contributing to a low global cluster cost, each node b in the decision state estimates its own fitness value $Fitness_b$ based on the parameters introduced above. At this point, the node waits using a delay which is proportional to the $1 - Fitness_b$ value. When the waiting time has elapsed, the node sends a *Membership Request* message to the clusterhead, informing it that it is willing to be included into the cluster. Now the clusterhead, based on the amount of additional resources needed for the cluster and the availability of resources of the candidate, can decide whether the node will be accepted as member. If accepted, the clusterhead includes the new node in a table with all members of the cluster. A message is sent back to the node confirming/refusing the entrance into the cluster. When receiving the response message, the requester changes its status accordingly. If accepted, this new status is broadcasted immediately in a message calling for new members (*Call Members*) to the neighborhood of *b*, starting the process again. When the cluster is complete, all additional receiving requests will be rejected.

3 Service Migration

In our fine-granular distributed RTOS *NanoOS*, services are distributed over the nodes of a cluster; the clusters being created as described above. The optimization goal here is to migrate services dynamically to such nodes that the global communication costs between services and application tasks requesting these services are minimized. Note that the requesting application tasks may reside on any nodes of a cluster. This problem again can be mapped onto an ACO problem. In our approach services are the equivalent of food sources, calls made by the requesters are the ants, and requesters are the formicaries. Wireless links constitute the paths which the ants can use for their walks. While the requests are being routed to the destination service, they leave pheromone on the nodes. The pheromone, on the other hand, evaporates over time. This rather classical solution is further enhanced to also consider the specific workload on the destination nodes of potential migrations. In addition geographically related paths are handled in such a way that they bundle attracting force into their direction. Details can be found in [12].
Although there is a wide range of middleware and virtual machine approaches, at this moment, the majority of operating systems for WSNs do not provide service assignment mechanisms. Given the fact that most task/service assignment mechanisms used in WSNs are online (deciding during run-time), code mobility is necessary for such approaches.

3.1 Formulation of the Problem

The objective is the positioning of the system services through *migration*, i.e. dynamically re-assigning the services to nodes in the system in order to reduce the communication overhead. This objective can be formulated as a formal optimization problem. The system is represented by two graphs. The first one is the *network (resource) graph* and the second one is the *processing thread (task/service) graph*.
Input: (1) A processing thread graph *T* modeling the communication requirements between the diverse processing threads of the OS and applications, and (2) a network graph *G*.
Constraints: The services and tasks assigned to node *v* do not request more resources than available at the node.
Costs: The cost of the multi-hop shortest path employing the virtual distance between nodes $u,v \in V$.
Goal: Minimize costs.

3.2 Basic Service Migration Heuristic

Using an analogy with the ant foraging behavior [6], the services are the equivalent of the food source. The calls made by the requesters are the agents (or ants) and the requesters are the formicaries. The wireless links form the pathway used by the ants. While the requests are being routed to the destination service, they leave pheromone on the nodes. The approach follows these policies:

Transfer policy: Each service is independent and may decide itself about starting a migration. The target of a service migration is every node with sufficient resources.

Selection policy: The selection policy is based on a threshold θ that is compared to the measure of the current communication overhead of the service. Each packet coming from the requester *r* to the service *s* carries the virtual distance traveled. Multiplying the size of the packet by the traveled distance, the communication cost of the packet is calculated.

Location policy: The location policy decides about which node should receive a migrating service.

Information policy: The heuristic uses almost only passive information gathering by means of pheromone tables. Any broadcasting or proactive information dissemination is avoided to save the scarce energy resources.

The general idea is to migrate the service to some node that relies in some requests flow (path) or near to it, in the direction of a requester. Each service has several flows coming from the diverse requesters. In order to determine which node should receive the service *s*, an explorer packet will be used. Its next hop is defined based on the pheromone value of the neighborhood and its final location will eventually be the target node for the migration of *s*.

In order to estimate the level of pheromone potentially caused by those flows if the service would migrate to the node being evaluated, the so called *potential pheromone* is introduced as the sum of all other pheromones related to the service *s*, coming from the neighbors not selected as next hop for the exploration packet. The main idea is to predict which situation would occur if the service would migrate to the current exploration packet position and which would be the next hop for a possible migration. This means, although the pheromone level from these flows would not appear to the exploration packet when far away from the node (*v*) hosting *s*, they should be considered when deciding the next exploration packet hop. After having selected possible candidate target nodes in the exploration phase, the settlement phase is responsible to find the appropriated node with enough resources to host the service. In the positive case, the service will migrate to the node. In the negative one, the neighborhood will be checked and, according to the actual situation of the neighborhood, a neighbor may be selected or the exploration packet may migrate to the last visited potential candidate, to search there for the final destination of the service *s*.

3.3 Enhanced Service Migration Heuristic

There is a problem caused by the greedy nature of the basic heuristic. The problem occurs when more than one nearby located requesters use the same service, but due to the employed routing algorithm, the requests are routed through different paths. The main idea of the improvement is not to migrate the service to the neighbor with the highest amount of requests (highest flow) as in the basic heuristic, but to the neighbor whose flow, in some part, is crossing nodes near to other flows requesting the same service. If the defined metric (virtual distance) has (geographical) norm properties, this will be equivalent to migrating the service to the geographical *direction* from where the highest amount of requests is coming.

4 Conclusions

The area of wireless sensor networks seems to be appropriate to get inspirations from biological systems. In a WSN the individual sensor node, its specific role, its specific communication links are of minor interest. What counts is the behavior of the entire network. It is not so important that always an optimal clustering could be achieved. However, it is essential that over time and under potentially hostile environmental influences a reasonable clustering is maintained in a robust manner. It is not so important that always shortest paths are used. However, it is important that a communication infrastructure with reasonable costs is maintained in a robust manner. It is not so important that services are always deployed at an optimal location. However, it is important that a reasonable service deployment is maintained in a robust and adaptive manner, even under dynamically changing situations. Nature is very efficient in providing such robust solutions. We demonstrated how some main demands of WSNs can be provided by means of biologically inspired techniques.

This contribution is a summary and combination of [10] and [12]. More detailed presentations of the approaches can be found there.

5 References

1. Amis, A., Prakash, R., Vuong, T., Huynh, D.: Max-min d-cluster formation in wireless ad hoc networks. In: Proc. of Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), Volume 1, pp. 32–41 (March 26–30, 2000)

2. Baker, D., Ephremides, A.: The architectural organization of a mobile radio network via a distributed algorithm. IEEE Transactions on Communications, 29 (11), pp. 1694–1701 (Nov 1981)

3. Baker, D.J., Ephremides, A., Flynn., J.A.: The design and simulation of a mobile radio network with distributed control. IEEE Journal on Selected Areas in Communications (SAC) 2 (1), pp. 226–237 (1984)

4. Bannerjee, S., Khuller, S.: A clustering scheme for hierarchical control in wireless networks. In: Proc. of IEEE INFOCOM, Anchorage, Alaska (2001)

5. Basagni, S., Chlamtac, I., Farago, A.: A generalized clustering algorithm for peer-to-peer networks. In: Proceedings of the Workshop on Algorithmic Aspects of Communication (Satellite workshop of ICALP), Bologna, Italy (1997)

6. Eric Bonabeau, Marco Dorigo, and Guy Theraulaz. Swarm Intelligence: From Natural to Artificial Systems. Oxford University Press, New York, NY (1999)

7. M. Dorigo, V. Maniezzo, and A. Clorni: The Ant System: Optimization by a Colony of Cooperating Agents. IEEE Transactions on Systems, Man, and Cybernetics, Volume 26, pp. 944–955 (1996)

8. Gerla, M., Tsai, J.T.C.: Multicluster, mobile, multimedia radio network. Wirel. Netw. 1 (3), pp. 255–265 (1995)

9. Heimfarth, Tales; Rettberg, Achim: NanoOS - Reconfigurable Operating System for Embedded Mobile Devices. In: International Workshop on Dependable Embedded Systems (WDES), Florianopolis, Brazil (October 17, 2004)

10. T. Heimfarth, P. Janacik, and F. J. Rammig: Self-Organizing Resource-Aware Clustering for Ad Hoc Networks. In: Proc. of Software Technologies for Embedded and Ubiquitous Systems (SEUS), LNCS Volume 4761, pp. 319–328, Springer (2007)

11. Heimfarth, T., Janacik, P.: Ant based heuristic for OS service distribution on ad hoc networks. In: 1st IFIP International Conference on Biologically Inspired Cooperative Computing (BICC), Volume 216, pp. 75–84, Springer (2006)

12. T. Heimfarth, P. Janacik: Experiments with Biologically-Inspired Methods for Service Assignment in Wireless Sensor Networks. In: IFIP Conference on Biologically Inspired Cooperative Computing (BICC), Springer (2008)

13. Heinzelman, W., Chandrakasan, A., Balakrishnan, H.: An application-specific protocol architecture for wireless microsensor networks. IEEE Transactions on Wireless Communications 1 (4), pp. 660–670 (October 2002)

14. Krishnan, R., Starobinski, D.: Message-efficient self-organization of wireless sensor networks. In: Proc. of IEEE Wireless Communications and Networking Conference, New Orleans, USA (2003)