Sensor-DMAC: Dynamic Topology Control for Wireless Sensor Networks

Stefano Basagni,* Alessio Carosi,† and Chiara Petrioli†
* Department of Electrical and Computer Engineering
Northeastern University
E-mail: basagni@ece.neu.edu
† Dipartimento di Informatica
Università di Roma "La Sapienza"
E-mail: {carosi, petrioli}@di.uniroma1.it

***Abstract*— We present Sensor-DMAC (S-DMAC), a new mechanism for topology control in wireless sensor networks. A (connected) fraction of the network nodes is efficiently selected to perform the network operation while all other nodes are switched to an energy-conserving "sleep mode." Based on DMAC, a clustering protocol enhanced with backbone formation, S-DMAC reduces to a minimum the overhead associated to node selection, backbone formation and maintenance, thus increasing the overall network lifetime. Ns2-based simulations have been performed to compare S-DMAC with DMAC and also with GAF, a recent solution for topology control in wireless sensor networks. These results show S-DMAC as more effective in providing connected and energy efficient routes of data from the sensor nodes to the network collection point (sink).**

I. INTRODUCTION

Applications for mobile ad hoc networks have been recently widened by the interest that both industry and academia are developing for ubiquitous and pervasive computing and communications. Large numbers of small, energy-constrained devices are internetworked according to the ad hoc paradigm to implement these major new forms of communication.

Wireless sensor networks (WSNs) are probably one of the most typical example of these networks. Among the main characteristics of WSNs are the large number of nodes and the constraints on node resources, especially on the energy availability at each node.

Several techniques have been proposed for limiting the energy consumption at the nodes, thus increasing the network lifetime. These techniques, often referred to as *topology control*, aim at taking advantage of the density of the network nodes for turning off (the radio interface of) some nodes (hence saving their power), leaving only a reduced number of them on to maintain network connectivity and to perform the network functions. These mechanisms are dynamic in the sense that there is a "rotation" of the nodes that are asleep and awake, so that the nodes that are awake are those with the higher residual energy. Energy saving solutions for topology control that work along these lines have been presented in [1]–[4].

The *Geographical Adaptive Fidelity* (GAF) [1] protocol exemplifies this kind of methodology. Based on nodes' position-awareness, GAF nodes are associated with a squared *virtual grid* of side r (the area of sensor deployment is divided into a number of these grids). The side of the grid is chosen as a function of the nodes transmission radius R so that any node in a grid has a link with any node in neighboring grids. In this sense, routing-wise all nodes in a grid are equivalent. This permits the definition of an efficient mechanism for the selection of the node in the grid that stays awake. Specifically, a leader node is dynamically elected among the sensors in each grid based on the nodes' residual energy: The higher the energy, the more suitable a node is to serve as grid leader. The election is performed by the nodes' according to a three-state protocol. Every node starts a *discovery* procedure by which it sets a timer to a time which, for example, is inversely proportional to its energy. Upon timer expiration the node sends out a discovery message and moves to the *awake* state. When in this state, a node periodically sends a discovery message carrying its residual energy-based rank. When a node which is either in the discovery or in the awake state receives a discovery message from a neighboring node with higher rank it enters the *sleep* state, where its radio interface is turned off, for lower energy consumption. In this way, a unique leader is elected in each grid. Leaders are the only nodes that remain awake to form the communication infrastructure (backbone). All other nodes stay asleep for a fraction of the time that the leader is expected to be able to perform its functions (sensing, data aggregation, data routing, etc.) Every time a node wakes up, it enters again the discovery phase which might take to its election as a new leader. In this way, a rotation of the leaders is obtained that leads to a balanced energy consumption at the nodes.

Although energy saving is obtained by keeping only the leaders awake, the relationship between r and R leads to the creation of small grids, which in turn implies that, even at moderate to high network densities, there is only a very limited number of nodes per grid, i.e., a very limited number of nodes that can be turned off. Experimental results show that, independent of the number $n \leq 300$ of the network nodes uniformly deployed in a square area of side 200 meters, when $R = 30$ meters the percentage of the grids with only one node (with respect to the non empty grids) is always $> 50\%$. In sparse networks ($n = 50$) this percentage is $> 80\%$. Finally, due to the needed small size of the grid, GAF may result in disconnected topologies, since it is very likely for a grid with one or more nodes to be adjacent to one or more empty grids. Thus, two non-neighboring non-empty grids might be connected or not depending on the particular leaders elected in each of them. In sparse networks most of the grids are empty (over 80% when $n = 50$; over 65% when $n = 100$). We have observed that for 42% of the simulation time "GAF networks" with 50 nodes get disconnected (i.e., sensed data cannot be routed to the data collection node, called the *sink*) even though the network topology (which comprises nodes that are awake and those that are asleep) is connected.

In this paper we propose a novel protocol for topology control in large wireless sensor networks. Differently from previous solutions, the density of the network does not affect the ability of providing a connected network of nodes that are awake. The protocol is based on the selection of a subset of the sensor nodes to build a connected backbone, which is made of backbone nodes and *gateways* nodes selected for interconnecting the backbone nodes. The backbone nodes are selected in such a way that every node not in the backbone (termed *ordinary node*) is a neighbor of at least a backbone node. The selection mechanism follows the basic rules for network clustering and backbone construction as implemented by the DMAC protocol presented in [5]. Upon starting the protocol operations, a DMAC node computes its *weight*, i.e., a real number > 0 which indicates how good that node is for being a backbone node. For instance, the weight could be computed based on the node's residual energy. This also implies that a node's weight changes in time, reflecting the changes in the node's status. The node then acquires knowledge of its neighbors' identity and weights and depending on the weights it decides whether to be a backbone node or not. This process is performed by having each node periodically sending out "hello" packets, which carry the node's identity, and its current weight. In particular, the node with the bigger weight in its (one hop) neighborhood will declare that it is going to be a backbone node. Consequently, all its neighbors are going to be ordinary nodes. Among the ordinary nodes, a few are chosen to act as gateways between backbone nodes that are at most two or three hops away. This is the essential (necessary and sufficient) rule that guarantees that the backbone composed of backbone nodes and gateways is connected [6]. Clearly, this is independent of the network density: As far as the network topology is connected, the backbone is connected as well. DMAC is able to accommodate network dynamics that corresponds to nodes mobility and to the arrival of new nodes at a later time. This feature is made possible by the continuous "monitoring" of a node's surroundings to determine the presence of new nodes: As soon as a new node is detected, relevant informations (identity, weight, etc.) are exchanged among the nodes, and suitable procedures are triggered to re-organize the backbone to include the new node while remaining connected. For instance, the death of a backbone node for energy depletion could bring to the election of one of its ordinary nodes, say A, as backbone node. According to DMAC, some of A's neighboring ordinary nodes will be affiliated to A if the backbone node to which they are currently affiliated has a weight smaller than A's. If A is neighbor of a backbone node B and B current weight is smaller of A's weight, B will resign as backbone node and the backbone re-organizes, including new nodes to stay connected [5]. The discovery of new nodes is performed by keeping all nodes transmitting hello packets continuously. Although costly, this mechanism allows prompt neighbor discovery and guarantees that the nodes in the backbone are always the fittest (e.g., those with the highest residual energy).

In this paper, we have exploited the DMAC basic rules for designing a new protocol, termed *Sensor-DMAC* (S-DMAC for short), enhancing DMAC with capabilities that make it suitable for the unique characteristics and requirements of sensor networks. The basic protocol design concept is to reduce to a minimum all transmissions between nodes related to nodes discovery and backbone construction and maintenance. Further energy saving is obtained by sending to sleep all the ordinary nodes, while keeping awake all the nodes that form the connected backbone.

In particular, S-DMAC implements the following mechanisms.

- Weight-based node selection uses nodes' residual energy.
- Overhead reduction for neighbor discovery at backbone set up and for backbone maintenance has been accomplished by limiting use of hello packets with respect to DMAC, and by minimizing backbone reorganization. The overhead due to backbone maintenance is kept as low as possible by changing the backbone structure only when strictly needed. In particular, the backbone is re-organized only when introducing a new batch of "fresh" sensor nodes with much higher energy of the current ones, or when backbone nodes die because of energy depletion. With respect to DMAC, an ordinary node will re-affiliate with a newly inserted backbone node only when the energy of the new backbone node exceeds the old one's energy of a predefined quantity h. This proves to effectively reduce the overhead of backbone maintenance, and more importantly, it keeps the changes local (differently from DMAC, which instead suffers from "chain reaction" re-clustering, where the change of status of a backbone node potentially triggers a network-wide backbone re-organization).
- In S-DMAC only the nodes in the backbone will be awake while all the ordinary nodes are asleep. We define a scheduling mechanism for a backbone node to send its neighboring ordinary nodes to sleep.
- Adaptivity to dynamic situations such as network *splits* (due to nodes leaving the network because of, e.g., energy depletion) and *joins* (when new nodes are added to the network). The mechanisms that we introduce to cope with splits and joins take into account that not all nodes are awake at the time new nodes enter the network or when the residual energy of a node is no longer enough to support the node's functions.

Ns2-based simulation results are shown (a) that demonstrate the energy-efficiency of S-DMAC in providing connected routes throughout the network, and (b) to compare the performance of S-DMAC, DMAC and GAF.

The paper is organized as follows. In the next session we describe the S-DMAC protocol in details. Section III describes our ns2 implementation of the three protocols, GAF, DMAC and S-DMAC as well as the scenarios considered for our experiments. In the same section we describe the metrics that we have considered to compare the protocols as well as the results of our simulations, which include a thorough discussion about the effectiveness of S-DMAC as a mechanism to prolong network longevity. Section IV concludes the paper.

II. SENSOR DMAC

In this section we describe the operation of S-DMAC at each network node. We describe the protocol as executed by the generic node v with weight $w_v > 0$, which here is a function of the node residual energy. (In case two nodes have the same weight, ties are broken by using the unique nodes' IDs.)

Upon entering the protocol, node v waits to receive packets from its neighboring nodes for a certain period of time τ which guarantees that all its neighbors will transmit a packet. During this time v listens in *promiscuous mode*, i.e., listens to all packets from any other nodes, even though v is not the intended destination of those packets. This allows v to become aware of the identity of its one hop neighbors, their weight, whether they will be part of the backbone or not, and, in the case they are ordinary nodes, to which backbone node they are affiliated. Based on the received packets, after the time τ has passed, v decides whether it is going to be a backbone node or not. This decision depends on the decision made by v's neighbors with bigger weight and by those neighbors with a weight bigger than v's weight minus a certain $h > 0$. More specifically, if there is no such neighbor that has decided to be a backbone node, v becomes a backbone node itself, and beacons this information to all its neighbors. Otherwise, it affiliates with the backbone node with the biggest weight, beaconing a corresponding packet. If v exits the "promiscuous phase" as a backbone node, and there is one of its neighbors u with weight $w_u < w_v - h$ that decided to be a backbone node as well before v did, then, according to the DMAC rules, u must resign. A backbone re-organization follows, which is performed according to the DMAC procedures. The parameter h is important for decreasing the likelihood of resignation/backbone re-organization of this kind. The higher the h the less likely v will force a smaller neighbor to resign. Intuitively, the parameter h implements the idea that a backbone re-organization is needed only when the new backbone node is really better than the current one. This, for instance, may happen when a batch of new, energy powerful nodes are added to the network. In this case makes perfect sense that the backbone is made up of the new nodes. In DMAC it is $h = 0$, which means that as far a $w_u < w_v$, node u must leave the backbone, which can trigger a costly backbone re-organization.

At network set up, once every node has exited the promiscuous phase, it has been assigned a role, which can be either backbone node or ordinary node. In the latter case, the ordinary node is affiliated to a backbone node. At this time a backbone node chooses its gateway nodes that interconnect it to its "adjacent backbone nodes" (i.e., to backbone nodes that are either two or three hops away). As mentioned, backbone nodes and gateways form a connected backbone for delivering sensed data to the sink. The choice of the gateways is performed similarly to the choice performed in DMAC. At this point, backbone nodes and gateways remain awake, while all ordinary nodes are sent to sleep for at most $t_{\max}$ seconds. An ordinary node wakes up for two possible reasons. One is when it has sensed data that need to be sent to the sink, in which case it immediately wakes up and transmits the data to its backbone node (to minimize the latency). The other reason is when its assigned asleep time $< t_{\max}$ expires. All ordinary nodes of a backbone node wake up basically at the same time. When this happens, the backbone node beacons out a packet containing backbone node ID, its current weight, the time its ordinary nodes have to stay asleep, and the list of the nodes that have to reply to this beacon. If a newcomer has been added to the network and it is in promiscuous mode, these beacons make the presence of the backbone node known to the newcomer. The nodes "polled" by the backbone node unicast a "presence" packet carrying the sender's ID, the backbone node to which it is affiliated and its current role and weight. An ordinary node is asked to send this packet every c wake-up periods. Sending a packet to its backbone node serves also the purpose of letting a newcomer node still in the promiscuous phase to know its neighboring gateways or ordinary nodes. If the newcomer stays in the promiscuous phase long enough, it can get ahold of all its one hop neighbors, their weights, roles and affiliations, which allows it to choose its role (which can trigger a backbone re-organization). The parameter c trades off between the time a newcomer has to stay in promiscuous mode (to discover its neighbors) and the energy consumption of the ordinary node (overhead expressed by the number of non-data packets it has to send to let its backbone node and newcomers know that it is still alive).

It is clear how there is very little exchange of control packets in building up the connected backbone of awake nodes. Problems arise when, due to energy depletion, one of the backbone nodes dies, and a backbone re-organization is in order. We have to consider multiple cases. In the unlikely situation when an ordinary node dies, its backbone node simply removes it from the list of its affiliated nodes. The communication of the node's death can be either directly via a dedicated packet (the node knows its energy is finishing up and spends the last little of it to inform its backbone node) or indirectly, when no packet is received by the backbone node when the node is supposed to send one. Similarly, a little more is to be done when a gateway node dies. The backbone nodes that used it as gateway must choose alternative gateways, if possible. It is possible to prove that, provided that the gateway death does not result in a disconnection of the network topology, the selection of new gateways is enough to maintain the backbone connected. The case when a backbone node dies is a bit more involved. We assume here that a backbone node can anticipate its own death by energy depletion, à la GAF [1]. Before this happens, when its ordinary nodes awake, the backbone node informs them to stay awake because a backbone re-organization has to occur. It also informs, via its gateways, the adjacent backbone nodes so that they can keep their affiliated ordinary nodes awake. Adjacent backbone nodes also exchange information about those ordinary nodes and gateways that are still alive (with their current weight). This limits the communication about the updates on the topology to when a backbone re-organization is needed. Assuming that the time for inter-backbone nodes communication is $< t_{\max}$ (which is quite an upper bound to reasonable sleep times), within at most $2t_{\max}$ all ordinary nodes affiliated to backbone nodes adjacent to a dying one are awake and are ready to perform the DMAC-like routines for the selection of new backbone nodes and gateways.[1]

III. Simulation Results

In order to assess the improvements achieved by S-DMAC in terms of low energy consumption and overhead in backbone construction, we have performed simulations of S-DMAC, GAF DMAC. The three protocols have been implemented in the VINT project simulator ns2 [7].

Our simulations refer to scenarios in which $n \leq 300$ stationary sensor nodes with transmission radius of 30m are randomly and uniformly scattered in a geographic area which is a square of side $L = 200$m. The sensor nodes initial energy is set to 50J. By keeping L fixed we are able to show the performance of the protocols on networks with increasing densities: The average nodal degrees varies from 6 (networks with 100 nodes) to 18 (when $n = 300$).

Up to 10 targets roam throughout the network according to the random way-point mobility model. When a sensor detects the presence of a target a corresponding data packet is routed to the sink, which is an unconstrained node randomly positioned in the deployment area.

The energy model used in our experiments closely follows the specifications of the TR 1000 radio transceiver from RF Monolithics [8]. The energy consumption corresponding to the different radio modes as well as the various protocol and simulation parameters are summarized in Table I.

The following is the meaning of some of the parameters. Parameter T_{target} denotes a sensor transmission data rate upon the detection of a target. Parameter "Start time" indicates the time by which nodes become operational. The "Energy threshold" parameter denotes the minimum residual energy required for the operation of a backbone node (BN) or of an ordinary node (ON), and $T_{\text{discovery}}$ controls how frequently discovery messages are transmitted while GAF is running.

[1] Inter-backbone node exchange of ordinary nodes' status could be pre-emptively done when a backbone node knows that it will die within a certain time. Via this exchange, and corresponding programmed awake of the ordinary nodes affiliated to adjacent backbone nodes, the delay of waiting for all nodes to awake can be rendered negligible.

TABLE I
Simulation parameters

Node parameters	
Tx range	30m
Sensing range	10m
Initial energy	50J
Tx power	14.88mW
Rx power	12.5mW
Idle power	12.36mW
Sleep power	0.016mW
Scenario parameters	
Nodes	≤ 300
Area	$200x200\text{m}^2$
T_{target}	1pck/20s
Pck data size	64byte
Targets	10
Movement	Random way-point
Random way-point parameters	
Speed	$(0, 1]$m/s
Pause	20s
S-DMAC parameters	
Start time	$[0, 40)$s
Energy threshold: BN	0.5J
Energy threshold: ON	0.1J
Promiscuous phase	40s
GAF parameters	
r	13m
Starting time	$(0, 1]$s
$T_{\text{discovery}}$	$[59, 60]$s (active)
$T_{\text{discovery}}$	$(0, 3]$s (discovery)

The metrics that we have investigated in this paper include the following averages:

A. The size of the generated backbones (this includes backbone nodes and gateways).
B. The percentage of simulation time in which the GAF backbone is disconnected (even if the topology induced by the nodes that are still operational is connected).
C. The number of bytes sent for control packets (backbone construction), per node.
D. The power consumption (per node).

All results refer to the case in which n sensor nodes enter the network at a time t, with t uniformly distributed in the interval $[0, 40)$s, and monitor the deployment area for all the network lifetime (i.e., till the time the first node dies for energy depletion). The results are obtained by averaging values obtained over 300 different (connected) topologies.

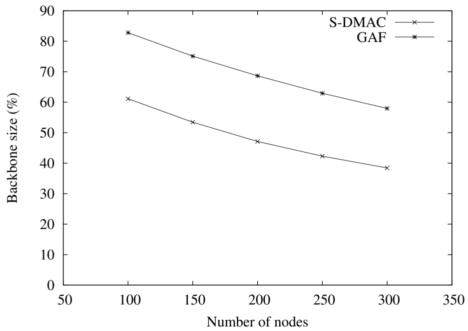

Fig. 1. Backbone size (%)

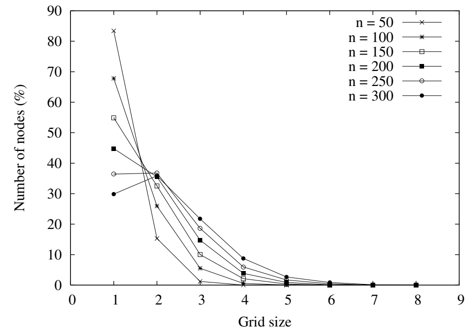

Fig. 2. Distribution of nodes in GAF grids

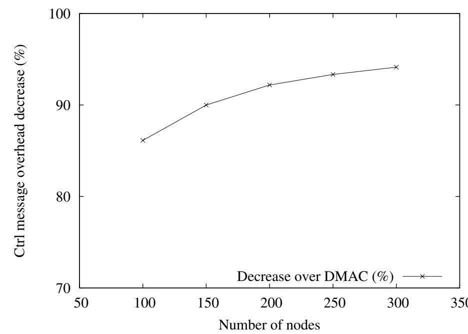

Fig. 3. Decrease in control packets per node (%) over DMAC

A. Backbone size

We first show that S-DMAC can effectively overcome typical GAF limits, i.e., the high percentage of nodes in the backbone and the possible network disconnections. Figure 1 shows the average backbone size of S-DMAC and GAF when varying the nodes density. We observe that S-DMAC produces a connected backbone considerably smaller than the GAF backbone. In networks with 300 nodes the backbone generated by the S-DMAC protocol is 44% smaller than the GAF backbone. This, in turn, induces major improvements in terms of energy consumption, as the nodes of the backbone are the only ones that are always awake (these nodes are the real bottlenecks in terms of energy consumption).

B. Backbone connectivity

In addition to reduced backbone size, S-DMAC has the significant advantage of resulting in no backbone disconnections during the network lifetime. Disconnections are instead typical in GAF: In networks with 50 nodes the backbone is disconnected 42.82% of the network lifetime. As expected, this percentage decreases as the nodal density increases: Disconnections occur for 28.06% of the simulated time in networks with 100 nodes, and less than 7% of the simulated time for bigger networks. However, even for moderately dense networks, the non-negligible probability of disconnections compromises reliable communications from the sensor nodes to the sink and vice versa.

As mentioned, the high probability of disconnection in the GAF backbone depends on the fact the grids are small, which leads to many empty grids. The low number of nodes per grid also motivates the large GAF backbone size. Figure 2 shows the percentage of nodes belonging to cells with a given grid size (defined as the number of nodes in the grid). In sparse topologies ($n = 50$) over 80% of the nodes belongs to grids of which they are the only residents. Even in denser topologies ($n \geq 250$), over half of the nodes can be found in grids with at most two nodes. Although decreasing with increasing values of n, the percentage of nodes which are leaders of a grid with no other nodes (and hence the percentage of nodes which are always awake) is never smaller than 30%.

C. Control packets overhead

We have quantified the advantages of using S-DMAC instead of DMAC in terms of reduced number of backbone re-organization and corresponding lower energy consumption. Figure 3 depicts the decrease (%) in number of control packets exchanged during the time the nodes enter the network in S-DMAC with respect to DMAC. This metric accounts for the cost of backbone re-organization which is induced by a node entering the network after one of its neighbors already decided to become a backbone node. If the later node has a better weight than the backbone node then, according to DMAC, a backbone re-organization is triggered (we recall that in DMAC is $h = 0$). Our results show that the control packet reduction achieved by using S-DMAC ranges from 86.13% in networks with 100 nodes up to 94.13% when $n = 300$.

D. Power consumption

A major limit of DMAC is the lack of a scheme to schedule the sleep times of the nodes. The fact that the transceivers at the nodes are always on leads to a considerably higher energy consumption in DMAC with respect to S-DMAC. This is clearly shown in Figure 4, which shows the decrease in energy consumption achieved by S-DMAC over DMAC (the average is per node). We observed that, in sparser topologies ($n = 100$) the power consumption of S-DMAC is 47.5% lower than the power consumption of DMAC. The decrease is more noticeable in dense topologies (i.e., those with 300 nodes) where an S-DMAC node consumes an average of 63.67% less energy than the corresponding DMAC node.

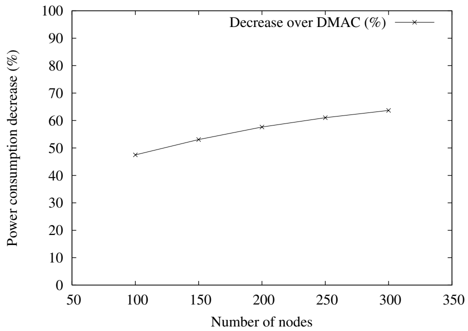

Fig. 4. Decrease in power consumption per node (%) over DMAC

IV. Conclusions

This paper introduced Sensor-DMAC (S-DMAC), a new mechanism for topology control in wireless sensor networks. S-DMAC is based on a dynamic clustering algorithm, DMAC, originally proposed for clustering ad hoc networks. The clustering mechanism has been used in S-DMAC for the dynamic selection of backbone nodes, which are the nodes that stay awake and coordinate the operations of all the other nodes. These nodes are either used for interconnection purposes or sent to sleep for energy conservation. Beyond being more energy efficient than DMAC, the proposed scheme significantly reduces the DMAC control overhead, keeping to a minimum hello-like (control) packets, and reducing the number of backbone re-organization. S-DMAC has been thoroughly described and simulation results have been presented that show a performance evaluation comparison between S-DMAC, DMAC and GAF, a previous solution for topology control in wireless sensor networks. We have observed that S-DMAC is effective in overcoming the major GAF drawbacks, namely, in providing a connected backbone with a smaller number of nodes. At the same time, being equipped with a schedule to send nodes to sleep, and having been designed to be a lightweight protocol (with respect to the use of control packets), S-DMAC outperforms DMAC in terms of node power consumption and increased network lifetime.

Overall, although preliminary with respect to a more thorough performance evaluation and comparison of the three protocols which is to come, the results of our experiments already demonstrate that S-DMAC effectively overcomes the limits of both GAF and DMAC. In this sense, S-DMAC provides a reliable, energy-efficient and low-overhead solution for topology control in self-organizing wireless sensor networks.

Acknowledgments

This work has been partially funded under the FIRB Virtual Immersive COMunications (FIRB VICOM) project and project IST-2001-34734 EYES (EnergY Efficient Sensor networks).

References

[1] Y. Hu, J. Heidemann, and D. Estrin, "Geography-informed energy conservation for ad hoc routing," in *Proceedings of the 7th ACM Annual International Conference on Mobile Computing and Networking*, Rome, Italy, July 16–21 2001, pp. 70–84.

[2] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, "SPAN: An energy efficient coordination algorithm for topology maintenance in ad hoc wireless networks," *ACM/Kluwer Wireless Networks*, 2002.

[3] F. Ye, G. Zhong, J. Cheng, S. Lu, and L. Zhang, "PEAS: A robust energy conserving protocol for long-lived sensor networks," in *Proceedings of the 23rd IEEE International Conference on Distributed Computer Systems, ICDCS 2003*, Providence, RI, May 19–22 2003, pp. 28–37.

[4] Y. Xu, S. Bien, Y. Mori, J. Heidemann, and D. Estrin, "Topology control protocols to conserve energy in wireless ad hoc networks," University of California, Los Angeles, Center for Embedded Networked Computing, Tech. Rep. 6, January 2003. [Online]. Available: http://www.isi.edu/~johnh/PAPERS/Xu03a.html

[5] S. Basagni, "Distributed clustering for ad hoc networks," in *Proceedings of the 1999 International Symposium on Parallel Architectures, Algorithms, and Networks (I-SPAN'99)*, A. Y. Zomaya, D. F. Hsu, O. Ibarra, S. Origuchi, D. Nassimi, and M. Palis, Eds. Perth/Fremantle, Australia: IEEE Computer Society, June 23–25 1999, pp. 310–315.

[6] I. Chlamtac and A. Faragó, "A new approach to the design and analysis of peer-to-peer mobile networks," *Wireless Networks*, vol. 5, no. 3, pp. 149–156, May 1999.

[7] The VINT Project, *The ns Manual*. http://www.isi.edu/nsnam/ns/, 2002.

[8] "ASH transceiver designer's guide," www.rfm.com, May 19 2004.