

A Performance Comparison of Protocols for Clustering and Backbone Formation in Large Scale Ad Hoc Networks

Stefano Basagni,* Michele Mastrogiovanni[†] 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: {mastrogiovanni,petrioli}@di.uniroma1.it

Abstract—This paper concerns the comparative performance evaluation of protocols for clustering and backbone formation in ad hoc networks characterized by a large number of resource-constrained nodes. A typical example of these networks are wireless sensor networks. We selected protocols that represent the main approaches to clustering and backbone formation for ad hoc networks. The DCA protocol represents those protocols whose backbone construction method is based on selecting nodes as clusterheads and then joining them to form a connected backbone. The algorithm proposed by Wu and Li has been chosen to exemplify those algorithms that build a connected backbone and then prune away redundant nodes. Finally, the algorithm by Wan et al. has been considered here for its more theoretical properties of producing a backbone with a constant approximation factor, linear time complexity and optimal message complexity. Extensive ns2-based simulation results show that simpler algorithms are rewarded with good performance with respect to most metric of interest which include per node energy consumption and message overhead. However, they tend to produce larger backbones. On the other side, more complex solutions produce smaller backbones at greater cost. In order to obtain a backbone reasonably small at reasonable cost we propose an enhancement of the DCA algorithm, termed DCA-S, which enriches the DCA backbone construction with a recently proposed and resource effective sparsification rule. DCA-S leads to a robust backbone close in size to that generated by the Wan et al. protocol without significantly degrading the performance in terms of all the other relevant metrics.

I. INTRODUCTION

Technological advances as well as the advent of 4G communications and of pervasive and ubiquitous computing have fostered a renewed interest on multi-hop (ad hoc) communications. In particular, there is an interest in self-organizing wireless multi-hop networks composed of a possibly very large number of nodes. These nodes can be either static or mobile, and are usually constrained as for the most critical resources, such as power and computation capabilities.

A typical example of this kind of networks is given by *wireless sensor networks* (WSNs) [1]. In this case, the well-known paradigm of ad hoc networking specializes to consider a higher number of nodes (in the thousands and more) that

are heavily resource-constrained. Rather than on mobility the emphasis is now on data transport from the sensors to other sensors or to specific data collection nodes (*sinks*). Sensor nodes are usually unreplaceable, and become unusable after failure or energy depletion. It is thus crucial to devise protocols for topology organization and routing that are scalable, energy conserving and increase the overall network longevity.

Among the solutions proposed for scaling down networks with a large number of nodes, *network clustering* is among the most investigated. The basic idea is that of grouping network nodes that are in physical proximity thereby providing the network with a logical organization which is smaller in scale, and hence simpler to manage.

Clustering is seen as the first step to provide the *flat* network with a *hierarchical* organization.¹

The subsequent *backbone* construction uses the clustering-induced hierarchy to form a communication structure that is functional in providing desirable properties such as minimizing communication (MAC and routing) overhead, choosing data aggregation points, increasing the probability of aggregating redundant data, and minimizing the overall power consumption.

Clustering for ad hoc networks has been investigated for over twenty years now. We describe the vast variety of solutions for ad hoc clustering and backbone formation in the next section. In general, to accommodate nodes mobility and for minimizing the clustering and backbone maintenance overhead, most of the proposed protocols end up generating a clustering and a corresponding backbone whose nodes form a *dominating set* of nodes in the flat network topology. This means that each node that is not in the backbone has at least a backbone node as its neighbor. The construction of such a set happens in different ways. For instance, approaches like the one proposed by Baker et al. [2] first construct an

¹ "Flat" is here used as opposed to "hierarchical." The flat network topology is the topology formed by all network nodes in which there is a link between two nodes if they are *neighbors*, i.e., if they are in each others communication range.

independent set of nodes, i.e., a set in which no two nodes are neighbors. In this way nodes from the set, also called the *clusterheads*, are well-spread throughout the network. Then, as indicated in [3], clusterheads are joined through *gateway* nodes to form a connected backbone that is a dominating set. If the flat network is connected, it is deterministically guaranteed that the backbone is connected as well. The concept of a communication *spine*, first introduced by Das et al. [4], is that of building a *connected dominating set* (CDS) directly, without selecting clusterheads and then joining them. The emphasis in this case is more on facilitating routing than on providing the network with a hierarchical organization.

The aim of this paper is to investigate and provide a thorough simulation-based comparison of clustering and backbone formation algorithms which are among the most representative solutions in this research area.

We selected three protocols for clustering and corresponding backbone construction, namely, the *Distributed Clustering Algorithm* (DCA) first presented in [5] and enriched with the backbone formation rule defined in [6], the backbone formation scheme presented in [7] and the two-phase algorithm presented in [8].

The DCA algorithm is the natural generalization of all those clustering protocols that are mainly concerned about selecting clusterheads that form a maximal independent set, like the above mentioned [2] and other work by Lin et al. that followed [9]. The backbone construction is not explicitly specified, and in order to obtain a connected backbone it is necessary (and sufficient) to establish paths among all clusterheads that are two and three hops away [6]. (Throughout the paper by DCA we will mean the DCA protocol and the corresponding backbone construction.)

The algorithm presented in [7], hereafter referred to as WuLi (after the combined names of the authors), represents the kind of protocols that builds a backbone directly. According to the authors this solution for building a CDS outperforms the solutions proposed by Das et al. Differently from the spine, which grows into a connected backbone from a non-connected dominating set, the WuLi algorithm finds a CDS, and then prunes some unnecessary nodes away from it.

Finally, the WAF protocol (so named here after the initials of the authors' last names) presented in [8] has been chosen for its more theoretical properties of producing a CDS with a constant approximation factor, linear time complexity and optimal, $\Theta(n \log n)$ message complexity. In this sense, the authors state that WAF outperforms the WuLi protocol and the spine solutions mentioned above, and we wanted to test this claim with respect to a realistic implementation of the protocols.

The contribution of this paper is twofold. We provide the first simulation-based performance comparison of protocols for clustering and backbone formation. In our ns2-based simulation we take into account characteristics of the network physical and MAC layers, and we make realistic implementation decisions at the protocol (network) layer. Secondly, we implement the chosen protocols as if they were executed in

wireless sensor networks, thus evaluating protocols that were initially proposed for the more general case of ad hoc networks in scenarios, with parameters and for metrics that are typical of WSNs. Specifically, we have considered the following metrics: a) the average time needed by each protocol to complete the clusterhead selection and backbone formation; b) the average overhead (in bytes) associated to each protocol operations; c) the average energy consumption per node; d) the average percentage of nodes in the backbone; e) the average route length over the backbone, and f) the backbone robustness, defined as the average number of backbone nodes failures leading to backbone disconnection.

As detailed in the performance comparison section, we observed that for large networks of energy constrained nodes, the simplicity of the protocol is rewarded in terms of performance. Due to its low complexity, the WuLi protocol outperforms the other two protocols in terms of duration, overhead, per node energy consumption and robustness, with DCA being a close runner up. A special note should be done here about the metric concerning the percentage of nodes in the backbone. Given the use of such a structure, primarily for data transport and routing, we consider backbones with a smaller number of nodes and links to be desirable. In this sense, the WAF protocol outperforms the DCA and the WuLi given that its specific aim is to build a small CDS. The WAF complexity, however, does not make it suitable for deployment in sensor networks. Therefore, we have looked for a solution that would fall in the middle between the extremely "slim," tree-like backbone produced by WAF and the more "opulent" DCA and WuLi backbones. To this aim, we have implemented a sparsification technique, originally presented in [10], whose task is to remove redundant nodes and links while maintaining connectivity. We have applied the sparsification technique to the DCA backbone, and we have observed that DCA-S (this is the name of the obtained variant) achieves what we were looking for: A reasonably small, yet robust, backbone at a reasonable cost.

The paper is organized as follows. In Section II we give a quite thorough overview of clustering and backbone formation for ad hoc and sensor networks. Section III describes the four protocols that we compare: DCA, DCA-S, WuLi and WAF. In Section IV we describe in details our simulation environment and we evaluate and comment on the performance of the four protocols with respect to the selected metrics of interest. Finally, Section V concludes the paper.

II. AD HOC CLUSTERING AND BACKBONE FORMATION

The notion of cluster organization has been investigated for ad hoc networks since their appearance. The first solutions aimed at partitioning the nodes into *clusters*, each with a *clusterhead* and some *ordinary nodes*, so that the clusterheads form an independent set, i.e., a set whose nodes are never neighbors among themselves. In [2], [11], a fully distributed Linked Cluster Architecture (LCA) is introduced mainly for hierarchical routing and to demonstrate the adaptability of the network to connectivity changes. The basic concept of LCA

is adopted and extended to define multi-level hierarchies for scalable ad hoc routing in [12]. With the advent of multimedia communications, the use of the cluster architecture for ad hoc network has been revisited by Gerla et al. [9], [13], [14]. In these latter works the emphasis is toward the allocation of resources, namely, bandwidth and channel, to support multimedia traffic in an ad hoc environment. These algorithms differ on the criterion for the selection of the clusterheads. For example, in [2], [9], [11] the choice of the clusterheads is based on the unique identifier (ID) associated to each node: the node with the lowest ID is selected as clusterhead, then the cluster is formed by that node and all its neighbors. The same procedure is repeated among the remaining nodes, until each node is assigned to a cluster. When the choice is based on the maximum degree (i.e., the maximum number of neighbors) of the nodes, the algorithm described in [14] is obtained.

As illustrated in the next section, the DCA algorithm generalizes these clustering protocols in that the choice of the clusterhead is performed based on a generic "weight" associated to a node. This attribute basically expresses how fit that node is to become a clusterhead. All these protocols produce a set of clusterheads that are independent, and criteria for joining them to form a connected backbone must be defined. In this paper we adopt the rule defined by Theorem 1 in [6]: In order to obtain a connected backbone it is necessary (and sufficient) to join all clusterheads that are at most three hops apart via intermediate nodes (called *gateways*). Clusterheads and gateways form the backbone. Different choices and definitions for the weights and the effects of the particular choice on the DCA and similar protocols have been investigated in [15], [16] and [17]. Clusterhead selection and backbone formation, although differently from the methods used in the mentioned solutions, are also the two fundamental steps of the WAF algorithm considered in this paper [8]. (Details are given in the next section.) Other protocols based on constructing an independent set of clusterheads and then joining them to form a backbone are defined in [18] where the choice of the clusterheads is performed based on the normalized link failure frequency and node mobility. Two rules are defined for joining the selected clusterheads. The first rule utilizes periodic global broadcast messages generated by every node but forwarded only by the clusterheads. Non receiving re-broadcasts from all neighboring nodes via its clusterhead makes an ordinary node aware of a disconnection. The ordinary node becomes a backbone node providing connectivity. The second rule is similar to the one used by the DCA algorithm to build up a backbone. In [19] and in [3] DCA is used as basic clustering and rules are then defined to limit the size of the clusters.

Once the nodes are partitioned into clusters, techniques are described on how to maintain the cluster organization in the presence of mobility (clustering maintenance). Since the emphasis in this paper is on the clustering and backbone set up, we do not consider mobility. It is worth mentioning, however, that mobility has been the driving design parameter for some clustering algorithm, such as the ones presented in [20].

Algorithms directly concerned about building a backbone

which is a connected dominating set have been presented in [4], [21] and [22]–[24]. The idea in this case is to seek for a dominating set and then grow it into a connected dominating set. The emphasis here is to build a routing structure, a connected *spine* that is adaptive to the mobility of the network nodes. Differently from the solutions mentioned above, which distribute and localize the greedy heuristic for finding a maximal (weight) independent set, these solutions are a distributed implementation of the Chvátal heuristic for finding a minimal set cover of the set of nodes. A similar approach is followed in [25] where a minimum set cover is built in a distributed and localized way: Nodes in the set cover are databases that contain routing information. A somewhat different approach is adopted in the WuLi protocol that we consider here [7]. Instead of constructing a dominating set and then to join its nodes to make it connected, a richer connected structure is built, and then redundant nodes are pruned away to obtain a smaller CDS. Several different pruning rules are investigated in [26]. The number of nodes in the CDS can be further reduced by using nodal degree and the nodes (GPS) coordinates, as proposed in [27].

More recent work for clustering and backbone set up is described for networks quite different from the general ad hoc model considered in this paper. In [28] some nodes are assumed to have "backbone capabilities" such as the physical radio capacity to communicate with other backbone nodes (i.e., the network nodes are assumed to be heterogeneous). The solution proposed in [29] constructs a CDS relying on all nodes having a common clock (time is slotted and nodes are synchronized to the slot).

We conclude this section by reviewing clustering and backbone formation protocols that have been proposed explicitly for wireless sensor networks. The main problem here is that of devising energy efficient techniques to transport data from the sensors to the sink. The overall goal, in general, is to increase the network lifetime. Hierarchical solutions like those provided by clustering and backbones appear to be viable for accomplishing this task.

Among the protocols that use clustering for network longevity one of the first is the Low-Energy Adaptive Clustering Hierarchy (LEACH) protocol presented in [30]. LEACH uses randomized rotation of the clusterheads to evenly distribute the energy load among the sensors in the network. In addition, when possible, data are compressed at the clusterhead to reduce the number of transmissions. A limitation of this scheme is that it requires all current clusterheads to be able to transmit directly to the sink (single-hop topologies). Improvements to the basic LEACH algorithms have been proposed by [31] and [32] where multi-layer LEACH-based clustering is proposed and the optimal number of clusterheads is analytically derived that minimizes the energy consumption throughout the network.

Clustering protocols have been proposed for networks in which some nodes are capable of long-haul communications. This is the case of the clustering proposed in [33] where special, more powerful nodes act as clusterheads for simpler

sensor nodes, and transmit the sensed data directly to the sink. Gerla and Xu [34] propose to send in swarms to collect data from the sensors, and to rely the data to the sink via intermediate swarms (multi-hop transport). Finally, the construction of a backbone of sensor nodes is considered in [35] where the nodes need to know their position (such as their GPS coordinates). This work is more along the lines of sensor network topology control (à la GAF [36]) rather than on hierarchical organization of WSNs as considered in this paper.

III. FOUR BACKBONE FORMATION PROTOCOLS

Our comparative analysis focuses on the following protocols: WuLi, WAF, DCA and DCA-S. As mentioned, these are all distributed, localized and deterministic algorithms for computing a backbone which is nothing but a connected dominating set (CDS). Ideally, these CDSs should be small, but we are also interested in other desirable properties which make them suitable for deployment in large ad hoc networks of resource constrained devices such as WSNs. The following discussion focuses on aspects of these algorithms that are relevant for the performance evaluation to follow.

WuLi is a very simple distributed procedure consisting of a few local rules, the execution of which creates the desired CDS. The simplest rule is the following: If a vertex v has two neighbors that are not in visibility range, then v enters a set C . It is straightforward that this rule eventually generates a CDS C (provided that the network is not a clique, a case that can be dealt with separately). The algorithm is extremely simple and efficient but, not surprisingly, tends to create very large CDSs. In [7] other local rules are discussed whose aim is to prune away unnecessary nodes. The two rules implemented here are the following. Rule 1: For every pair of nodes u and v in C the one with the smaller ID, say v , can be removed from C if v and all its neighbors are covered by u (i.e., they are u 's neighbors). Rule 2: Assume nodes u , v and w are in C , and assume that v 's ID is the smaller of the three. Assume also that u and w are neighbors of v . If each neighbor of v is covered by u or w then v can be removed from C .

In the worst case, the ratio between the size of the computed CDS and the optimum (a minimum CDS) can be as bad as $\Theta(n)$. (This also holds for the kinds of graphs that interest us here, namely, unit disk graphs [37] which model multi-hop networks). This is however a worst case scenario and one could hope in better average performances in practice.

The great virtue of this algorithm resides in its simplicity which greatly eases its implementation. Neighboring nodes simply need to exchange their neighbors list. Based on this sole information a node decides whether to enter C or not. Nodes then communicate their membership to C and, based on the information they receive from their neighbors, apply Rule 1 and Rule 2 to make the final decision on whether they will be part of the backbone or not.

The low complexity of WuLi implementation, as showed in the next section, translates into low communication and

computation cost. What our experiments also show is that, as expected, WuLi produces quite a large CDS.

In order to describe the other algorithms it is appropriate to introduce some general ideas on which these algorithms, and many others, are based. The basic approach to compute a small CDS is to (a) compute a (hopefully) small dominating set, and then (b) to connect it up [3], [8], [10], [23], [38]. For step (a) there are two strategies. One is to give a distributed implementation of the best algorithm (in terms of size) for dominating set computation. This is the well-known greedy heuristic for set cover (see for instance [39]). This greedy strategy repeatedly selects the vertex that dominates the highest number of non-dominated nodes, puts it into the solution, and proceeds. It is well known that this simple heuristic gives the best possible approximation of the optimum that can be obtained in polynomial-time, unless $P = NP$ [39]. (A distributed implementation of this heuristic is the approach followed in [23].)

A different approach to (a) is to compute a maximal independent set. An *independent* set is a collection of vertices such that no two of them are connected by an edge. If an independent set is maximal then it is also a dominating set. This is the approach taken by the DCA and WAF protocols.

Note that a maximal independent set in general tends to be large. For the kind of geometric graphs we consider here, however, the size of a maximal independent set is at most 5 times that of a smallest dominating set, and it is therefore a good approximation of the optimum.

Once the dominating set is computed the task is to connect it to form a backbone. The approaches used by DCA and WAF are of opposite nature. To illustrate them, we introduce the following definitions. Let $G = (V, E)$ be our network and let $D \subseteq V$ be the dominating set computed in step (a). To connect D consider the following *auxiliary graph* $H(D)$ with vertex set D . Two vertices u and v in D are connected by an edge if their distance in G is at most 3. It is straightforward that H is connected. Now, every edge uv of H corresponds to a path p_{uv} of length at most 3 in G , i.e., a path with at most two vertices. Let $P(uv)$ denote the vertices of p_{uv} . To obtain a CDS C from D simply add all such vertices to C , i.e., $C = D \cup (\bigcup_{uv \in H} P(uv))$.

In a synchronous, distributed setting computing C in this fashion takes constant-time and even in an asynchronous environment we expect the computation and communication cost to be low. Nodes in D simply need to gather two-hops neighbors information. Specifically, they have to be made aware about which neighbors of their neighbors are in D or are dominated by a node in D . Based on this information every node in D selects gateways to interconnect with other nodes in D which are at most three hops away.

The selected gateways are part of the backbone. DCA uses a maximal independent set computation for step (a) and the above procedure for step (b). In this case interconnecting gateways are selected according to a weight based criterion.

Step (a) of the DCA algorithm is executed at each node in such a way that a node v decides its own role (clusterhead

or ordinary node) depending solely on the decision of its neighbors with bigger weights. Thus, initially, only those nodes with the bigger weight in their neighborhood will broadcast a message to their neighbors stating that they will be clusterheads. On receiving one or more of this “clusterhead” messages, a node v will decide to join the cluster of the neighboring clusterhead with the biggest weight. If no node with bigger weight has sent such a message (thus indicating that it is going to join some other cluster as an ordinary node), then v will send a clusterhead message. Eventually, the set of clusterheads is a maximal independent set and hence a dominating set as well. Except for the initial routine, executed by each node to start the protocol operations, the algorithm is message driven: a specific procedure will be executed at a node depending on the reception of the corresponding message. Only two types of messages are used: CH(v), used by a node v to make its neighbors aware that it is going to be a clusterhead, and JOIN(v, u), with which a node v communicates to its neighbors that it will be part of the cluster of clusterhead u .

The simplicity of this algorithm translates into low computation and communication cost and makes it a very appealing candidate for computing a dominating set.

The second approach to connect D is much more economical in terms of size, but it is computationally expensive. The idea is to connect D by computing a spanning tree T of H , and to augment D with those vertices that correspond to edges of T . That is, $C = D \cup (\bigcup_{uv \in T} P(uv))$.

The way this can be implemented is by first electing a leader r among the nodes, which is going to be the root of T . Then T is constructed via an r -started flooding of a tree construction message m . While m travels through the network each node selects as its parent in T the node from which it received m first. Computing a tree for step (b) is basically the solution adopted by WAF [8].

Distributed leader election is *enormously* expensive in practice, and exhibits a very low degree of parallelism. Computing a spanning tree or electing a leader are highly centralized, non local tasks that require coordination, and hence message exchanges, between far away nodes in the network. The solution adopted in WAF for leader election, for example, requires nodes to be progressively joined to form a connected *fragment* with a leader. At the beginning of the algorithm each node is an isolate fragment of which it is the leader. As leader election progresses adjacent fragments merge into one, and the leader of the bigger fragment is selected to lead the newly formed one. Every time two fragments merge into one, the information on the new fragment size and the ID of its leader needs to be propagated to the fragment members and to the nodes in adjacent fragments. For this reason, although these algorithms are quite good in terms of CDS size, we expect them to be extremely expensive in terms of time and communication cost. Our simulations of WAF confirm this in full.

The last approach to step (b) tries to find a compromise between these two extremes. The algorithm described in [10] finds this compromise by departing significantly from previous

approaches. The basic idea is the following. In order to connect D up, find a subgraph S of H that is connected and sparse. By “sparse” we mean that $|S| = O(|D|)$. To clarify, a spanning tree connects D up with $|D| - 1$ edges, but we would be satisfied to do the same with, say, $5|D|$ edges. To create S the authors make use of an old lemma of the great late Mathematician Paul Erdős. Roughly this lemma says that if a graph does not have small cycles then it must have few edges. Therefore to sparsify H it suffices to destroy all small cycles. The lemma by Erdős ensures that by destroying all cycles of length $O(\log |H|)$ the remaining graph will have $O(|H|)$ many edges (a cycle is destroyed if at least one of its edges is deleted). The crux is how to destroy all small cycles while maintaining connectivity. In a distributed fashion this can be done by the following, amazingly simple procedure. Let us call a cycle *short* if it has length $c \log n$, for some fixed c (in practice, $c = 2$), and assume that each edge has a unique identifier. Then the algorithm is simply this: The lowest ID edge of every short cycle is deleted. Clearly, this destroys all short cycles. The perhaps surprising thing is that it leaves a connected subgraph S of size $O(|D|)$ that connects all vertices of D . The resulting CDS is then $C = D \cup (\bigcup_{uv \in S} P(uv))$.

In practice in our implementation we limited ourselves to destroy only triangles and squares from the DCA backbone. We termed the DCA protocol enhanced with this “sparsification” the DCA-S protocol.

As we shall see the DCA-S policy strikes a good compromise between the more complex WAF approach on one side and the lighter WuLi and DCA protocols in that it finds a small CDS at a very reasonable cost.

IV. PROTOCOLS COMPARISON

A thorough comparative performance evaluation of the clustering protocols described in Section III has been performed by means of simulations. The four protocols have been implemented in the VINT project network simulator (ns2) [40] and their performance have been compared to analyze the costs associated to clustering and backbone formation, as well as the properties of the resulting backbone. In particular, we have considered the following metrics: a) the average time needed by each protocol to complete the clusterhead selection and backbone formation; b) the average overhead (in bytes) associated to each protocol operations; c) the average energy consumption per node; d) the average percentage of nodes in the backbone; e) the average route length over the backbone, and f) the backbone robustness, defined as the average number of backbone nodes failures leading to backbone disconnection. Our ns2 implementation is based on the CMU wireless extension, i.e., on the IEEE 802.11 MAC here modified to take sensor nodes characteristics into account.

Our simulations refer to static scenarios in which n wireless nodes with maximum transmission radius of 30 meters are randomly and uniformly scattered in a geographic square area of side L . We make the assumption that two nodes are neighbors if and only if their Euclidean distance is ≤ 30 m. We call *visibility graphs* the topologies generated by drawing

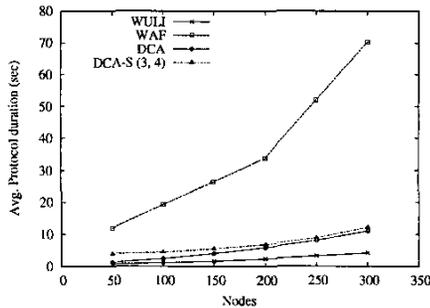


Fig. 1. Duration of the protocols.

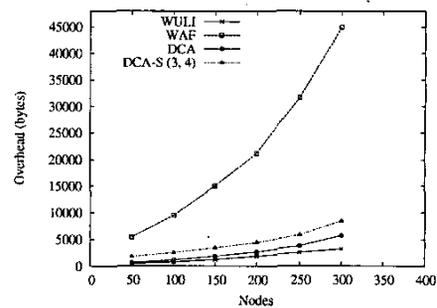


Fig. 2. Number of protocol bytes.

an edge between each pair of devices that are neighbors. Each device has an initial residual energy equal to 1J. The power consumed while transmitting, receiving, and while in asleep mode are equal to 24mW, 14.4mW and 0.015mW, respectively. (These values are from the actual energy model of the sensor prototypes developed within the IST Energy Efficient Sensor Networks, EYES, project [41].) The power consumed while in the idle state has been set equal to the power consumed while asleep. This corresponds to comparing the protocols performance under an ideal schedule of the nodes awake time and asleep time which keeps the nodes on only when needed.

In our simulations the number of nodes n has been assigned the values 50, 100, 150, 200, 250 and 300, while L has been set to 200m. This allows us to test our protocol on increasingly dense networks, from (moderately) sparse networks, with average degree equal to 3.5 to dense networks ($n = 300$) where the average degree is around 20.

All the sensors start the protocol at the same time and run the clustering and backbone formation algorithms to form a hierarchical multi-hop ad hoc network. Results presented in this section were obtained by running the protocols on 300 connected visibility graphs.

A. Protocol duration

Figure 1 depicts the average time needed by the WuLi, WAF, DCA and the DCA-S protocols to complete their operations. WuLi is the fastest protocol. This is due to its low complexity, the reduced number of information that each node needs to exchange with its neighbors, and the possibility to exchange this information in parallel.

The DCA (and DCA-S) protocol(s) also show reasonably good performance, requiring a time which ranges from 1.28s (3.97s) when $n = 50$ to 11s (12s) when $n = 300$. The increase (up to 150%) in duration with respect to the WuLi protocol is due to the fact that the DCA operations require a node to wait for all its neighbors with biggest weight to communicate their role before it can decide its own. This creates chains of dependencies that add time to the overall protocol operations. Also, once clusterheads are elected, the messages exchanged between potential gateways and clusterheads in order to select the gateways and to form the backbone may require significant information exchange and some extra time (though much less

than for clusterheads selection). At $n = 100$ ($n = 200$) 1.5s (3.9s) are required on average to partition the nodes into clusters, while 0.91s (1.9s) are needed for joining clusters into a connected backbone.

It might appear counter-intuitive that DCA-S—which is the DCA enriched with a phase where nodes detect whether they are part of three or four hop cycles—has a duration similar to that of DCA. This is due to the fact that no dependency chains slow down the communication of information during this phase and that only a few nodes participate in this phase. Since results are obtained averaging over the time required by all the nodes to complete the protocol, the extra delay required by some nodes to implement the “S” part of the protocol is balanced by many nodes quitting the protocol immediately after the backbone formation. Only the nodes that are selected as part of the backbone in DCA need to proceed to the extra phase in which cycles are identified and broken. By computing the average time needed by the DCA backbone nodes to complete the “S” phase we observed that this phase is quite short: No longer than 4s (on average).

WAF is by far the slowest protocol: 70s are needed to produce the connected backbone. This is due to the non-trivial complexity of the first phase of the protocol (leader election) that requires a very large number of messages to be exchanged to progressively join network fragments and to identify a common leader. The distributed implementation of this phase is time and overhead demanding, and constitutes a real bottleneck for the protocol adoption in practical scenarios.

B. Byte overhead

As a measure of the message complexity (overhead) of the clustering and backbone formation protocols, we have counted the average number of bytes of protocol packets transmitted by a node for performing the protocol operation. Figure 2 summarizes the results for the four protocols.

As expected, WAF is by far the more demanding protocol in terms of message overhead, requiring up to 14 times the protocol bytes of simpler protocols such as DCA and WuLi. This is again due to the overly high complexity of the WAF leader election phase which requires information to be propagated to the fragment members and to nodes in adjacent fragments every time two fragments are merged into a new

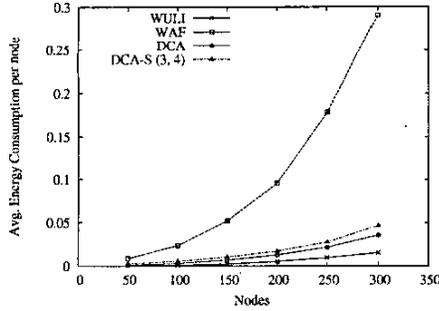


Fig. 3. Energy consumption (per node).

one. When $n = 200$ the leader election phase accounts for 86% of the total overhead. This phase introduces complexity in the protocol implementation (a very high overhead, fast increasing with n) and demands significant time and energy consumption.

DCA, DCA-S and WuLi show much better performance due to their simpler operations and to the limited number of information that need to be exchanged. WuLi is the best performing protocol, as neighboring nodes only need to exchange neighbor lists. DCA requires up to twice as many bytes to be transmitted. The DCA overhead is mostly due to the backbone formation phase, in which neighbors lists are exchanged (and nodes communicate to their clusterheads the ID, weight, role, and clusterhead ID of each of its neighbors). As n increases the overhead associated to this phase gets higher and higher with respect to the clusterhead election phase: The overhead for backbone formation is 35% higher than that for clusterhead election at $n = 100$ but it already increases to 175% with respect to the overhead of clusterhead election when $n = 200$. DCA-S shows a moderate increase in overhead with respect to DCA, since the number of nodes involved in the “S” phase of the protocol is quite limited.

C. Energy consumption

Per node energy consumption is an important metric as it points out potential severe limitations of clustering applicability. If the cost of clustering and backbone formation is high in terms of overhead and energy consumption, then clustering reorganization may be a non negligible burden for an energy constrained network. On the other hand, clustering reorganization such as clusterhead and gateway rotation is needed to prevent nodes with energy-demanding roles from depleting their energy, possibly impairing network operations.

Figure 3 depicts the average energy per node consumed for performing the selected protocols. (Results are expressed in Joule.)

As expected, the energy consumption curves follow the trend observed for the overhead closely, given the dependency of energy consumption on the number of bytes transmitted and received by each node. WAF is, again, the most energy consuming protocol. In networks with 300 nodes each WAF node consumes, on average, 0.3J for constructing the backbone.

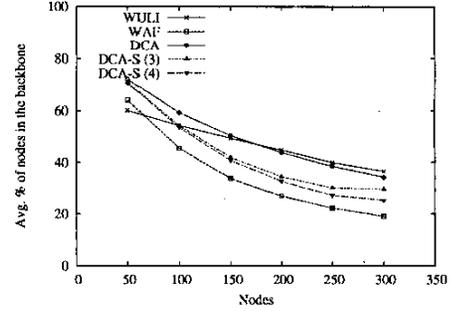


Fig. 4. Percentage of backbone nodes.

DCA-S, DCA and WuLi nodes requires considerably less: 0.046J, 0.035J and 0.014J, respectively. DCA-S consumes the same energy independently of whether it breaks triangles or squares. This is due to the fact that, in order to identify h -hop loops, $h = 3, 4$, each node sends/receives messages to/from its neighbors distant at most 2 hops. Hence there is no difference in terms of messages exchanged between the case $h = 3$ and the case $h = 4$.

D. Backbone size

The two major advantages of adopting a hierarchical organization in energy-constrained WSNs concern routing overhead minimization and the natural choice of points of aggregation of the sensed data (eliminating redundancy by means of data fusion at the clusterheads results in a smaller number of packets traveling in the network, energy saving and higher throughput). A smaller backbone improves network performance not just with respect to routing overhead but also because a smaller number of nodes need to be awake for data transport. This is important in energy-constrained systems such as sensor networks in which nodes need to spend as much time as possible in asleep mode (the wireless interface is not operational and the energy consumption is much lower) to reduce energy consumption, thus maximizing network lifetime. Given the low transmission range of these networks (from 10 to 30 meters) other topology control mechanisms such as power control are of lesser help.

Our claim is that a small backbone combined with “role rotation” (to balance the energy consumption associated to being part of the backbone) is the solution for efficient data transport in wireless sensor networks. We have investigated this metric for the selected protocols. The average percentage of nodes in the backbones is shown in Figure 4.

All the curves show that the percentage of nodes in the backbone decreases as n increases: Growing network density implies bigger cluster size and a wider choice of gateways. This is particularly evident for the DCA, where gateways interconnecting clusterheads that are two hops apart are more frequent than those that interconnect three-hops away clusterheads as n increases.

WAF generates a thin “tree-like” backbone, which implies a much lower percentage of nodes in the backbone with respect

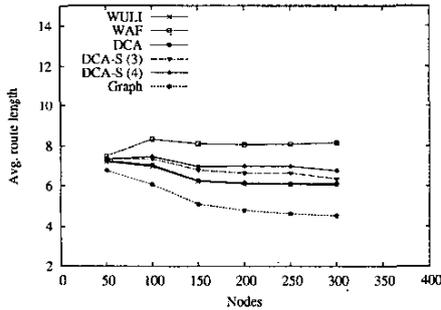


Fig. 5. Shortest path length.

to DCA and WuLi. For networks with 300 nodes the 19% of the nodes are in the WAF backbone, while almost twice as much (34%) form the DCA and WuLi backbones. The DCA-S curves prove the effectiveness of the sparsification rules applied to the DCA backbone. Not only the average degree of the nodes in the backbone and the average number of links in the backbone topology is significantly smaller, but in DCA-S a considerably lower number of nodes enter the backbone for interconnecting clusterheads, as clearly depicted in the figure. The percentage of nodes in DCA-S backbones when triangles (squares) are broken ranges from 70% (70%) when $n = 50$ to 29.55% (25%) when $n = 300$, resulting in decreased backbone size ranging from 3% to 26% over the DCA backbone.

E. Shortest path length

Independently of the particular routing protocol to be used in the sensor network, the average length of the shortest route is a metric that gives a measure of how well a routing protocol can perform. Figure 5 shows the increase in the average shortest paths length with respect to the same metric in the visibility graphs.

An increase in the average shortest path length is expected, as clustering might force nodes that are neighbors in the visibility graph to communicate through their common clusterhead (if they belong to the same cluster), or through a possibly long backbone route in case they belong to different clusters. This is reflected in the curves depicted in Figure 5 which show an average increase in the route lengths ranging from 7% ($n = 50$) to 34.7% ($n = 300$) in the topology generated by the DCA and WuLi protocols vs. the average shortest path length in the visibility graph. DCA-S produces sparser topologies with respect to those generated by DCA, thus resulting in longer average route lengths (up to 9%). The WAF protocol is the one resulting in longer routes since it produces a backbone which is very close to a tree. At $n = 300$ the average route length of WAF is 33.4% longer than DCA's.

F. Backbone robustness

Figure 6 shows how many backbone node failures (due to malfunctioning nodes or nodes that have depleted their energy) have to occur before the backbone gets disconnected. The backbone is defined as the subgraph of the visibility

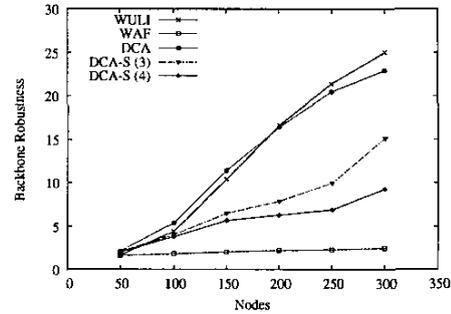


Fig. 6. Backbone robustness

graph spanned by the backbone nodes. This metric provides an indirect measure on how long the network will be operational before requiring a backbone recomputation.

As expected, the backbone robustness increases with the network density. In case of very sparse visibility graphs, a single node failure may lead to disconnections even in the flat network. As the network density increases, it gets more and more likely that when a backbone node failure occurs, the routes going through it can be rerouted to different backbone nodes.

The denser the backbone, the more resilient it is to node failures. Dense and heavily meshed backbones such as those resulting by the WuLi and DCA protocols may tolerate up to 25 failures in network with 300 nodes.

The thin WAF backbone, instead, cannot survive many failures: the failure of two nodes is on average more than enough to disconnect it. Quite interestingly, although DCA-S generates backbones almost as thin as WAF backbones, its resilience to nodes failures is remarkable. In dense networks ($n = 300$) up to 10 nodes can fail before the DCA-S backbone becomes disconnected.

V. CONCLUSIONS AND FURTHER RESEARCH

In this paper we presented a comparative ns2-based performance evaluation aimed at assessing the suitability of four ad hoc clustering protocols for scenarios such as those of wireless sensor networks. The selected protocols, DCA, WuLi, WAF and DCA-S, are representatives of the major clustering and backbone formation approaches proposed in the literature and range from independent set based approaches to spine based backbone formation protocols.

The obtained simulation results show that approaches with nice theoretical features, such as WAF, may hardly be applicable in practice due to the complexity of their operations. Simple solutions such as WuLi and DCA can be good starting points to develop a "light" and effective solution for sensor networks. In particular, the WuLi protocol is the best performing protocol in terms of all the relevant metrics except for the size of the backbone. When interested in limiting the backbone size for energy saving a good compromise appears to be the DCA-S protocol. DCA-S leads to a robust backbone close in size to that generated by WAF, without significantly

degrading the performance in terms of all the other relevant metrics.

The results obtained in this paper refer to the “basic” WuLi protocol. Dai and Wu have recently proposed a generalization of the two original WuLi rules [42]. The new rule appears to be effective in further reducing the size of the generated CDS. In our future work we intend to investigate the pros and cons of the enhancement to WuLi as well as the impact of different values for h in DCA-S. We finally intend to assess the impact on the protocol performance of different ways of choosing nodes’ weights, such as the degree-based approach suggested in [27].

ACKNOWLEDGMENTS

The authors wish to thank the anonymous reviewers for their valuable comments and suggestions.

Alessandro Panconesi is also thanked for fruitful discussions about CDS construction.

This work has been partially funded under the project IST-2001-34734 EYES (Energy Efficient Sensor networks).

REFERENCES

- [1] I. F. Akyildiz, W. Su, Y. Sanakarasubramaniam, and E. Cayirci, “Wireless sensor networks: A survey,” *Computer Networks*, vol. 38, no. 4, pp. 393–422, March 2002.
- [2] D. J. Baker, A. Ephremides, and J. A. Flynn, “The design and simulation of a mobile radio network with distributed control,” *IEEE Journal on Selected Areas in Communications*, vol. SAC-2, no. 1, pp. 226–237, January 1984.
- [3] S. Basagni, D. Turgut, and S. K. Das, “Mobility-adaptive protocols for managing large ad hoc networks,” in *Proceedings of the IEEE International Conference on Communications, ICC 2001*, vol. 5, Helsinki, Finland, June 11–14 2001, pp. 1539–1543.
- [4] B. Das and V. Bharghavan, “Routing in ad-hoc networks using minimum connected dominating sets,” in *1997 IEEE International Conference on Communications, ICC’97*, Montreal, Quebec, Canada, 8–12 June 1997, pp. 376–380.
- [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] J. Wu and H. Li, “On calculating connected dominating sets for efficient routing in ad hoc wireless networks,” *Telecommunication Systems, Special Issue on Mobile Computing and Wireless Networks*, vol. 18, no. 1/3, pp. 13–36, September 2001.
- [8] P.-J. Wan, K. M. Alzoubi, and O. Frieder, “Distributed construction of connected dominating sets in wireless ad hoc networks,” *ACM/Kluwer Mobile Networks and Applications, MONET*, vol. 9, no. 2, pp. 141–149, April 2004.
- [9] C. R. Lin and M. Gerla, “Adaptive clustering for mobile wireless networks,” *Journal on Selected Areas in Communications*, vol. 15, no. 7, pp. 1265–1275, September 1997.
- [10] D. Dubhashi, A. Mei, A. Panconesi, J. Radhakrishnan, and A. Srinivasan, “Fast distributed algorithms for (weakly) connected dominating sets and linear-size skeletons,” in *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, Baltimore, MD, 12–14 January 2003, pp. 717–724.
- [11] A. Ephremides, J. E. Wieselthier, and D. J. Baker, “A design concept for reliable mobile radio networks with frequency hopping signaling,” *Proceedings of the IEEE*, vol. 75, no. 1, pp. 56–73, January 1987.
- [12] E. M. Belding-Royer, “Multi-level hierarchies for scalable ad hoc routing,” *ACM/Kluwer Wireless Networks*, vol. 9, no. 5, pp. 461–478, September 2003.
- [13] M. Gerla and C. R. Lin, “Multimedia transport in multihop dynamic packet radio networks,” in *Proceedings of International Conference on Network Protocols*, Tokyo, Japan, 7–10 November 1995, pp. 209–216.
- [14] M. Gerla and J. T.-C. Tsai, “Multicenter, mobile, multimedia radio network,” *Wireless Networks*, vol. 1, no. 3, pp. 255–265, 1995.
- [15] C. Bettstetter and R. Krauser, “Scenario-based stability analysis of the distributed mobility-adaptive clustering (DMAC) algorithm,” in *Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc 2001*, Long Beach, CA, October 4–5 2001, pp. 232–241.
- [16] M. Chatterjee, S. K. Das, and D. Turgut, “An on-demand weighted clustering algorithm (WCA) for ad hoc networks,” in *Proceedings of IEEE Globecom 2000*, vol. 3, San Francisco, CA, November 27–December 1 2000, pp. 1697–1701.
- [17] F. Garcia Nocetti, J. Solano Gonzales, and I. Stojmenovic, “Connectivity based k -hop clustering in wireless networks,” *Telecommunication Systems*, vol. 22, no. 1–4, pp. 205–220, 2003.
- [18] U. C. Kozat, G. Kondylis, B. Ryu, and M. K. Marina, “Virtual dynamic backbone for mobile ad hoc networks,” in *Proceedings of the IEEE International Conference on Communications (ICC 2001)*, vol. 1, Helsinki, Finland, June 11–14 2001, pp. 250–255.
- [19] T. Nieberg, P. Havinga, and J. Hurink, “Size-controlled dynamic clustering in mobile wireless sensor networks,” in *Proceedings of the 2004 Communication Networks and Distributed Systems Modeling and Simulation Conference, (CSDN’04)*, San Diego, CA, January 18–21 2004.
- [20] A. B. McDonald and T. Znati, “A mobility-based framework for adaptive clustering in wireless ad hoc networks,” *IEEE Journal on Selected Areas in Communications, Special Issue on Wireless Ad Hoc Networks*, vol. 17, no. 8, pp. 1466–1487, August 1999.
- [21] P. Sinha, R. Sivakumar, and V. Bharghavan, “Enhancing ad hoc routing with dynamic virtual infrastructures,” in *Proceedings of IEEE Infocom 2001*, vol. 3, Anchorage, AK, April 22–26 2001, pp. 1763–1762.
- [22] R. Sivakumar, B. Das, and V. Bharghavan, “The clade vertebrata: Spines and routing in ad hoc networks,” in *Proceedings of the IEEE Symposium on Computer Communications (ISCC’98)*, Athens, Greece, June 30–July 2 1998.
- [23] R. Sivakumar, B. Das, and B. V., “Spine-based routing in ad hoc networks,” *ACM/Baltzer Cluster Computing Journal*, vol. 1, pp. 237–248, November 1998, special Issue on Mobile Computing.
- [24] R. Sivakumar, P. Sinha, and V. Bharghavan, “CEDAR: A core-extraction distributed ad hoc routing algorithm,” *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, pp. 1454–1465, August 1999.
- [25] B. Liang and Z. Haas, “Virtual backbone generation and maintenance in ad hoc network mobility management,” in *Proceedings of the 19th IEEE Infocom*, vol. 3, Tel Aviv, Israel, March 26–30 2000, pp. 1293–1302.
- [26] J. Wu, F. Dai, M. Gao, and I. Stojmenovic, “On calculating power-aware connected dominating sets for efficient routing in ad hoc wireless networks,” *Journal of Communications and Networks*, vol. 4, no. 1, pp. 1–12, March 2002.
- [27] I. Stojmenovic, M. Seddigh, and J. Zunic, “Dominating sets and neighbors elimination-based broadcasting algorithms in wireless networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 13, no. 1, pp. 14–25, January 2002.
- [28] K. Xu, X. Hong, and M. Gerla, “An ad hoc network with mobile backbones,” in *Proceedings of the IEEE International Conference on Communications (ICC 2002)*, vol. 5, New York, NY, April 28–May 2 2002, pp. 3138–3143.
- [29] L. Bao and J. J. Garcia-Luna-Aceves, “Topology management in ad hoc networks,” in *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc 2003*, Annapolis, MD, June 1–3 2003, pp. 129–140.
- [30] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-efficient communication protocol for wireless microsensor networks,” in *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, HICSS 2000*, Maui, HA, January 4–7 2000, pp. 3005–3014.
- [31] S. Bandyopadhyay and E. J. Coyle, “An energy efficient hierarchical clustering algorithm for wireless sensor networks,” in *Proceedings of the 22nd IEEE Infocom 2003*, vol. 3, San Francisco, March 31–April 3 2003, pp. 1713–1723.
- [32] —, “Minimizing communication costs in hierarchically clustered networks of wireless sensors,” in *Proceedings of the IEEE Wireless*

- Communications and Networking Conference (WCNC 2003)*, vol. 2, New Orleans, LA, March 16–20 2003, pp. 1274–1279.
- [33] G. Gupta and M. Younis, "Fault-tolerant clustering of wireless sensor networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC 2003)*, vol. 3, New Orleans, LA, March 16–20 2003, pp. 1579–1584.
- [34] M. Gerla and K. Xu, "Multimedia streaming in large-scale sensor networks with mobile swarms," *ACM SIGMOD Record*, vol. 32, no. 4, pp. 72–76, December 2003.
- [35] P. Santi and J. Simon, "Silence is golden with high probability: Maintaining a connected backbone in wireless sensor networks," in *Proc. of the 1st IEEE European Workshop on Wireless Sensor Networks, EWSN 2004, LNCS 2920*, Berlin, Germany, January 19–21 2004, pp. 106–121.
- [36] 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.
- [37] B. N. Clark, C. J. Colburn, and D. S. Johnson, "Unit disk graphs," *Discrete Mathematics*, vol. 86, pp. 165–167, 1990.
- [38] S. Guha and S. Khuller, "Approximation algorithms for connected dominating sets," *Algorithmica*, vol. 20, no. 4, pp. 374–387, April 1998.
- [39] S. Rajagopalan and V. V. Vazirani, "Primal-dual RNC approximation algorithms for set cover and covering integer programs," *SIAM Journal on Computing*, vol. 28, no. 2, pp. 525–540, 1998.
- [40] The VINT Project, *The ns Manual*. <http://www.isi.edu/nsnam/ns/>, 2002.
- [41] P. J. M. Havinga, S. Etalle, H. Karl, C. Petrioli, M. Zorzi, H. Kip, and T. Lentsch, "EYES—energy efficient sensor networks," in *Proceedings of the 4th IFIP TC6/WG6.8 International Conference on Personal Wireless Communications (PWC) LNCS 2775, M. Conti, S. Giordano, E. Gregori and S. Olariu (eds.)*, Venezia, Italy, September 23–25 2003, pp. 198–201.
- [42] F. Dai and J. Wu, "An extended localized algorithms for connected dominating set formation in ad hoc wireless networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 15, no. 10, October 2004.