

# Localized Protocols for Ad Hoc Clustering and Backbone Formation: A Performance Comparison

Stefano Basagni  
Department of Electrical and Computer Engineering  
Northeastern University  
E-mail: basagni@ece.neu.edu

Michele Mastrogiovanni, Alessandro Panconesi and Chiara Petrioli  
Dipartimento di Informatica  
Università di Roma “La Sapienza”  
E-mail: {mastrogiovanni,ale,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. Our aim is twofold. We provide the first simulation-based detailed investigation of techniques for clustering and backbone formation that are among the most representative of this area of ad hoc research. Furthermore, we delve into the nature of the selected protocols to assess the effects of their “degree of localization” on protocol performance, i.e., how being able to execute the protocol based only on local information improves the overall performance. Extensive ns2-based simulation results show that highly localized protocols are rewarded with good performance with respect to all metrics of interest which include protocol duration, energy consumption, message overhead, route length and backbone size.

## 1 Introduction

The upcoming generation of wireless systems, the fourth, will comprise a menagerie of technologies organized both in infrastructure-based architectures (wireless LANs, cellular networks) and in multi-hop, self-organizing networks (ad hoc networks) characterized by the lack of any fixed support. Computation and communication will be pervading every living space, and beyond. Nodes of a wide range of dimensions will be located where it is not viable for humans to proceed (e.g., disaster areas and outer space). This new kind of highly heterogeneous networks is made possible by those technological advancements for which a possibly very large number of nodes are able to self-organize themselves into a multi-hop, ad hoc network. This network may then be internetworked with more traditional wired networks such as the conventional telephone backbone or the general Internet.

A typical example of large multi-hop networks is given by *wireless sensor networks* (WSNs) [3,20]. Here 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 irreplaceable, 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 that increase the overall network longevity.

Whether a general mobile ad hoc network or a large sensor network with small, static nodes, the major challenge that multi-hop networks protocol designers have to face concerns the dynamic nature of the network topology and its increasing size. Nodes and links are added and removed, unpredictably, at all times.

In ad hoc networks, the mobility of the nodes greatly affects link quality and availability, thus changing the topology of the network continuously. Mobility is only one of the factors for the varying topology of sensor networks. The “death” of a sensor node for energy depletion or for other kinds of failure imposes changes in the network topology *and* size. For these reasons it is basically impossible to apply any kind of *centralized* approach to network communications: It is simply infeasible to gather information at a central unit, process the obtained input and then spread the results back to the nodes efficiently. Therefore, solutions for multi-hop network protocols at all levels must be *distributed*, namely, able to run at multiple nodes, and, more importantly, *localized*: protocols are performed at the nodes based on information on the local topology (typically, only a few of the surrounding nodes).

Of 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 size, and hence simpler to manage.

Clustering is seen as the first step to provide the *flat* network with a *hierarchical* organization.<sup>1</sup> The subsequent *backbone* construction uses the clustering-induced hierarchy to form a communication infrastructure that is functional in providing desirable properties such as minimizing communication 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 survey the vast variety of distributed and localized solutions for ad hoc clustering and backbone formation in the next section. In general, to accommodate node 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

---

<sup>1</sup> “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. We will also refer to the flat network as the network *visibility graph*.

happens in different ways. For instance, approaches like those proposed by Baker et al. [5] first construct an *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 [14], 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*, introduced by Das et al. [26], 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 the performance and provide a thorough simulation-based comparison of clustering and backbone formation algorithms which are among the most representative solutions of this research area.

We selected several protocols for clustering and corresponding backbone construction. In particular, we considered three major classes of protocols, which we distinguish based on their *degree of localization*. The degree of localization of a protocol is a measure of how much the decision of a node about being part of the backbone or not depends on nodes that are possibly several hops away from it. The computation is always localized, i.e., each node requires the knowledge of nodes at most  $h$  hops away (usually  $h \leq 3$ ), but, depending on the definition of the protocol, some nodes might have to wait a certain time before being able of making their decision, and this time depends on the decisions of other nodes, possibly not in their proximity.

The first class that we consider contains protocols that exhibit a high degree of localization. As soon as a node has collected information about its surrounding topology, it is able to decide whether it will be part of the backbone or not. The only information it needs to wait for is the identity (and maybe some other property) of the node in its ( $h$  hop) neighborhood. Protocols like “WuLi” first presented by Wu and Li in [62, 63] and enhanced in [25], and the *MultiPoint Relay* (MRP) protocol introduced in [1] belong to this class. The main idea behind these approaches is to start by building a very rich CDS and then apply local *rules* for removing nodes and corresponding links: Connectivity and dominance are maintained while the size of the CDS is remarkably reduced.

We then consider those protocols that implement a distributed version of the well-known heuristics for finding an independent set of nodes which is maximal, and a (non necessarily connected) dominating set which is minimal. As a representative of the first type of protocols we consider the *Distributed Clustering Algorithm* (DCA) [11], which builds on solutions like [5] and [44]. Being maximal, the independent set produced by DCA is also a minimal dominating set. A distributed implementation of the sequential, centralized greedy algorithm for finding a minimal dominating set is presented in [43]. The execution time of this algorithm

can be proven polynomial in the number  $n$  of the network nodes. For this reason, in this paper we consider its randomized version, the *Local Randomized Greedy* (LRG) [39], which generates a dominating set in poly-logarithmic time (with high probability).

Both DCA and LRG produce minimal dominating sets that are not necessarily connected. (They surely are not in the case of DCA). 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 according to the connectivity result established in [24]. (Throughout the paper by DCA and LRG we will mean the DCA and LRG protocols which include the extra step of backbone construction.)

The degree of localization shown by DCA and LRG is limited with respect to that of WuLi and MRP because of the very nature of the sequential (greedy) algorithms of which they are the faithful distributed implementation. These algorithms proceed in rounds. At each round a node is made capable of deciding whether to be part of the dominating set or not based on decisions made by neighboring nodes at previous rounds. If the nodes whose decision a node depends upon have not decided what to be yet, the node itself have to keep waiting. The degree of localization is thus limited for the possible presence of “chains of dependency” that prevent a node to make a faster decision. Consider for example, the clusterhead selection part of the DCA algorithm when run over a topology made up of a long chain of nodes with decreasing IDs. If nodes are elected to the dominating set based on their ID (bigger IDs beat smaller ones), according to DCA, before nodes with smaller IDs can make their decision, they have to wait for the decision of all their neighbors with bigger IDs. In extreme cases, like a chain, nodes with small IDs wait a time proportional to  $n$ . Our experiments show that the degree of localization does not only affect backbone construction time, but also the protocol message complexity.

Finally, the WAF protocol (so named here after the initials of the authors’ last names) presented in [58] 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. Moreover, the WAF protocol basically produces a tree-like backbone and hence a structure with a small number of nodes, which is considered desirable in certain networking scenarios. The authors state that WAF outperforms the WuLi protocol and the spine solutions mentioned above, and we decided to test this claim over a realistic implementation of the protocols. The degree of localization showed by WAF is definitely low because in order to build the tree upon which the backbone is based a root for the tree must be selected. This leader election process, although distributed as needed, is inherently sequential, and hence a node must wait for a leader to be elected and a tree to be constructed before being able to decide whether it is going to be part of the backbone or not.

The contributions of this paper are multifold. First, we provide a simulation-based performance comparison of protocols for clustering and backbone formation, with a particular emphasis on the impact of the

degree of localization over critical performance metrics. 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. We implement the selected protocols choosing as node and network parameters those that are typical of WSNs. Specifically, the metrics we consider include (averages): *a*) the time needed by each protocol to complete clusterhead selection and backbone formation; *b*) the overhead (in bytes, per node) associated to the protocol operations; *c*) the energy consumption per node; *d*) the percentage of nodes in the backbone; *e*) the route length over the backbone, and *f*) the backbone robustness, defined as the number of backbone nodes failures leading to either backbone disconnection or to the backbone no longer being able to cover all the other network nodes.

Secondly, we have looked at ways of improving the performance of the protocols. To this purpose we have considered extensions of the selected protocols that could affect their performance positively. The WuLi protocols has been evaluated with the “pruning” rule defined by Stojmenovic et al. in [55]. An enhancement of MPR is proposed by Wu in [60], where a smaller forward node set (a CDS) is generated basically at no additional cost. We notice that these modifications to the protocols, although algorithmically simple, are extremely effective in improving protocol performance.

Motivated by the observation that backbones are primarily used for data transport and routing, we consider backbones with a smaller number of nodes and links to be desirable. As expected, we have observed that the WAF protocol outperforms the WuLi, MPR (and their extensions), DCA and LRG since its specific aim is to build a small CDS. The WAF complexity, however, as well as its low degree of localization, does not make it suitable for deployment in sensor networks. Therefore, we have looked for a solution that would fall in between the extremely “slim,” tree-like backbone produced by WAF and the richer backbones generated by the other protocols. To this aim, we have implemented a sparsification technique, originally presented in [27], whose task is to remove redundant nodes and links while maintaining connectivity. We have applied this sparsification technique to the DCA and LRG backbones, and we have observed that DCA-S and LRG-S (these are the names of the obtained variants) achieve what we were looking for: A reasonably small, yet robust, backbone at a reasonable cost.

We have simulated the selected protocols according to their degree of localization. Specifically, we have performed three large sets of experiments.

The first set concerns protocols such as WuLi and MPR, which exhibit a high degree of localization. The performance of these protocols and of their implemented variants has been carefully scrutinized in order to select the protocol version that show the overall better performance.

The second batch of experiments concerns those protocols with a lower degree of localization such as DCA and LRG. Similarly to what we did for WuLi and MPR we have investigated their performance in

order to determine which among the two protocols and their sparsified variants is best for clustering and backbone formation in networks with resource constrained nodes.

Finally, the two selected protocols from the first two classes have been compared with the protocol that produces the thinner backbone, namely, WAF. As detailed in the performance comparison section, we observe that for networks like WSNs, the simplicity of a protocol and a high degree of localization are rewarded in terms of performance. Although WAF is basically unbeatable in producing backbones made up of a very small fraction of the network nodes, its low degree of localization and the complexity of its leader election phase make it basically unusable for WSN. We found that a well-sparsified version of the DCA implements the needed compromise between backbone size and robustness. With respect to these two metrics, this protocol outperform the best among WuLi, MPR and their variants. For all the other relevant metrics, the higher degree of localization of solution à la WuLi makes a remarkable difference. In particular, when coupled with the degree-based pruning rule defined in [55], WuLi builds decently sized backbones quickly, and spending the lowest energy and overhead among all the investigated backbone formation solutions.

The paper is organized as follows. In Section 2 we give a quite thorough overview of clustering and backbone formation protocol for ad hoc and sensor networks. Section 3 describes the selected protocols, WuLi, MPR, DCA, LRG and WAF and their extensions, introduced according to their degree of localization. In Section 4 we describe in details our simulation environment and we evaluate and interpret the performance of the protocols with respect to the selected metrics of interest. Finally, Section 5 concludes the paper.

## 2 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 [5] and [28], 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 [15]. With the advent of multimedia communications, the use of the cluster architecture for ad hoc network has been revisited by Gerla et al. [30,31,44]. In these latter works the emphasis is toward the allocation of resources, namely, bandwidth and channel, to support multimedia traffic in the ad hoc environment. These algorithms differ on the criterion for the selection of the clusterheads. For example, in [5,28,44] 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 [31] 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 use the rule defined by Theorem 1 in [24]: 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 [18], [22] and [29]. The effects of mobility on the basic clustering produced by the DCA, as well as methods for reducing role changes (protocol overhead), has been considered in [10, 11, 16, 17]. Clusterhead selection and backbone formation, although different from the methods used in the mentioned solutions, are also the two fundamental steps of the WAF algorithm considered in this paper [58].

Other protocols based on constructing an independent set of clusterheads and then joining them to form a backbone are defined in [41] 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 [49] and in [14] 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 [46]. Wang and Olariu discuss the problem of clustering maintenance at length in [59], where they also present a tree-based clustering protocol based on the properties of diameter-2 graphs.

Algorithms directly concerned about building a backbone which is a connected dominating set have been presented in [26], [51] and [52–54]. 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 [43] 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 [63]. 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 [61]. In this paper we consider the most recent formulation for the pruning rules as presented in [25]. The number of nodes in the CDS can be further reduced by using nodal degree and the nodes (GPS) coordinates, as proposed in [55].

Most of the clustering protocol mentioned so far generate clusters of diameter  $\leq 2$ : The clusterhead always dominates its cluster members. There have been advocates for larger, possibly overlapping clusters. For instance, [42] describes routing for dynamic networks (such as ad hoc networks) which is based on overlapping  $k$ -clusters. A  $k$ -cluster is made up of a group of nodes mutually reachable by a path of length  $k \geq 1$  (1-clusters are cliques). Clustering construction and maintenance in face of node mobility is presented, as well as the corresponding ad hoc routing. A clusterhead election protocol, with corresponding cluster formation is described in [4]. The focus in this paper is to efficiently build disjoint clusters in which each node is at most  $d \geq 1$  hops away from its clusterhead. The network is clustered in a number of rounds which is proportional to  $d$ , which favorably compares to most of previous solutions when  $d$  is small. Finally, the mentioned clustering protocol presented in [46] produces clustering of variable diameter. In this case the diameter depends on the degree of mobility of the nodes: The more the nodes move, the smaller the clusters (easier to maintain), and vice-versa.

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 [64] 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 [9] 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, as demonstrated in [12, 13, 45]. Rather than belonging to one of the general classes of protocols for clustering and backbone formation considered in this paper, papers on clustering for WSNs are often specific for a given scenario of application, and, as mentioned, are designed to achieve given



desirable goals, such as prolonged network lifetime, improved tracking, etc.

Among the protocols that use clustering for increasing network longevity one of the first is the Low-Energy Adaptive Clustering Hierarchy (LEACH) protocol presented in [37]. 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 [6] and [7] 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. An alternative method for selecting clusterheads for the LEACH-like model is presented in [57]. Particle swarm optimization (PSO) is used for dividing the network into clusters. A clusterhead is then selected in each cluster based on its mean distance from all the other nodes in the cluster.

Localized clustering for wireless sensor networks has been proposed by Chan et al. in [21]. The Algorithm for Clustering Establishment (ACE) has the sensor nodes iteratively talking to each other until a clustering with clusterheads and followers (the ordinary nodes) is formed where the clusters are quite uniform in size and mostly non-overlapping. This minimizes the number of clusterheads. No backbone formation among the clusterheads is described. A tree-based clustering algorithm for sensor networks is presented in [8]. A selected node starts the BFS-based process of building a spanning tree of the network topology. Clusters are then formed by those nodes whose sub-tree size exceeds a certain threshold. The protocol is quite message intensive, and there is no explicit description of backbone formation.

Clustering protocols have been proposed for networks in which some nodes are capable of long-haul communications (heterogeneous networks). 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. The authors have further explored their idea in [34, 35, 40, 65, 66] and [2]. Gerla and Xu [32] 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). Clustering for sensor networks of heterogeneous nodes is also explored in [23]. Clusters are formed dynamically, in response to the detection of specific events (e.g., acoustic sensing of a roaming target). Only a certain type of node can be clusterhead, and methods are presented for selecting the more appropriate clusterhead for target identification and reporting. The detailed description of how the sensed data is delivered to the sink is not among the concerns of this paper. A comparison among sensor networks with homogeneous nodes and with heterogeneous nodes is presented in [48], both for topologies “single-hop” (à la LEACH) and for the multi-hop case. The same authors make the case for heterogeneous sensor networks in [47]. The authors also propose M-LEACH, a variation of LEACH where intra-cluster

communications are multi-hop, instead of having ordinary nodes directly accessing their clusterhead.

Finally, the construction of a backbone of sensor nodes is considered in [50] 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 [38]) rather than on hierarchical organization of WSNs as considered in this paper. Further references on CDS construction in sensor and ad hoc networks can be found in [19].

### 3 Protocols Description and Implementation

Our comparative analysis focuses on the following protocols: WuLi, MPR (with their extensions), DCA and DCA-S, LRG and LRG-S, and finally WAF. As mentioned, these are all distributed and localized protocols for computing a backbone which is nothing but a connected dominating set (CDS). The algorithms are quite different, especially concerning the degree of localization that they exhibit. In the following description, and in the performance evaluation section that follows this important aspect of the considered protocols has been the leading point of view from which we have presented and compared them.

WuLi [62] 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 set  $C$  which is a CDS (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. Therefore, the authors discuss other local rules whose aim is to prune away unnecessary nodes.

The original protocol presents two pruning rules. 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 smallest of the three. Assume also that  $u$  and  $w$  are neighbors of  $v$  and are in each other transmission range. If each neighbor of  $v$  is covered by  $u$  or  $w$  then  $v$  can be removed from  $C$ .

The version of the WuLi protocol that we consider in this paper is the one presented in [25] where Rule 1 and Rule 2 are generalized into the following Rule  $k$ : Assume that vertex  $u \in C$ , and that a subset  $S$  of  $k$  neighbors of node  $u$  is such that (a) the sub-graph spanned by  $S$  is connected, (b)  $S$  is contained in  $C$ , (c) each vertex in  $S$  has ID larger than  $u$ , and (d) each neighbor of  $u$  is covered by the nodes in  $S$ . Then,  $u$  can be removed from  $C$ . Being the most general, this last rule prunes away the largest number of nodes. In the following we will denote with  $WuLi(i)$  this version of the WuLi protocol where  $k = i$ .

We implemented WuLi in the following way. Every node exchanges its neighbor list with its own neighbors. Based on this simple and local exchange of information a node can decide whether it is in  $C$  or not (high

degree of localization) and communicates this information to its neighbors. Once it is aware of which of its neighbors belong to  $C$ , a node  $u$  in  $C$  locally executes Rule  $k$  to decide whether it will be included in the final backbone or not.

From an (ns2) implementation point of view, this protocol uses three messages. The first is for generating the rich CDS, and the following two are used for communicating to the neighbors its membership in  $C$  and its final decision. All messages are broadcast by each node to all its neighbors. Each transmission has an associated timer. If the timer expires and the node  $u$  has not received the corresponding message from all its neighbors, the  $u$  sends a unicast message to all those neighbors from which it did not hear, asking for that message re-transmission. The first message is re-sent in broadcast (again), while the last two, more critical, are instead re-transmitted using a unicast packet. Time-outs have a fixed length (0.7s for most of the implementations), which is increased with a random jitter (chosen in  $[0, 0.3]$ s) to decrease the likelihood of collisions. (This basic time-out settings are typical of the implementation of the other protocols as well.)

Stojmenovic et al. [55] propose an effective strategy for improving the performance of the WuLi protocol. The idea is that there is no need for a node in  $C$  to exchange its membership information with its neighbors. Each node can locally determine whether it belongs to  $C$  or not. If this is the case, Rule  $k$  can be applied to the entire two-hop “neighbor sub-graph,” independently of whether the neighbors belong to  $C$  or not (as nodes do not communicate such decision, there is no way for a node to know which of its neighbors are in  $C$ ). This idea spares a node the exchange of a message (the second message in the implementation of WuLi) without affecting the “pruning power” of Rule  $k$ . Moreover, the nodes are now sorted according to their degree rather than their IDs (which are now used only for breaking possible ties). This positively affects the sparsification efficacy of Rule  $k$ .

The “Stojmenovic” implementation of the WuLi algorithm uses only two messages: The first allows the nodes to collect information about their two-hop neighborhood, and the second is used by a node for informing its neighbors about its final decision.

A high degree of localization is also shown by the simple and elegant protocol proposed in [1]: The *MultiPoint Relays*-based construction of a CDS, termed MPR. The basic idea is to have each node building a local CDS the subgraph induced by its two-hop neighbors. Nodes in the local CDS form the sets of multipoint relays. This name is reminiscent of the fact that this protocol was initially proposed for broadcasting: The broadcast message is forwarded only by multipoint relay nodes. Finding the multipoint relays is quite fast, given the evident high degree of localization of the corresponding protocol. There is an initial exchange of messages via which a node is made aware of its two-hop neighborhood (this happens exactly as for the WuLi algorithm). After this phase, a node  $u$  locally selects a set  $C_u$  of its neighbors as its multipoint relays by using the following simple greedy algorithm. Step 1: A neighbor  $v$  of  $u$  is inserted in  $C_u$  if there is a two-hop

neighbor of  $u$  covered only by  $v$ . Step2: A neighbor  $v$  of  $u$  is inserted in  $C_u$  if  $v$  covers the largest number of uncovered two-hop neighbors of  $u$  (this step is iterated till all two-hop neighbors are covered).

It is clear that node  $u$  does not have to wait but for the information about its two-hop neighbors. The set  $C$  of all  $C_u$ s is a CDS of the entire network: A node belongs to  $C$  if it belongs to one of the  $C_u$ s. As in the case of the WuLi protocol,  $C$  can be pretty rich. Furthermore, it is not necessarily connected. Therefore, the authors define a couple of rules to ensure connectivity. Rule  $a$  stipulates that a node  $u$  enters the final backbone  $B$  if it has the smallest ID among all its (one-hop) neighbors. Rule  $b$  states that  $v$  enters in  $B$  also if it is a multipoint relay of its neighbor with the smallest ID. The node terminates the protocol by communicating its final decision to all its neighbors. The resulting set  $B$  is proven to be a CDS, as required. Note that these rules also prune away redundant nodes. We implemented this protocol by using three messages: The first is for exchanging the list of neighbors (broadcast), the second is used by a node  $u$  to communicate to a neighbor  $v$  for which  $u$  is the neighbor with the smallest ID that  $v \in C_u$  (unicast), and the final one is used by a node to make every neighbor aware of its final decision (broadcast).

Jie Wu [60] has noticed that some nodes selected by Rule  $a$  are not essential for a CDS. Moreover, the basic greedy algorithm does not take advantage of Rule  $b$ . Hence, in [60] two enhancements to the MPR original scheme are proposed for obtaining a smaller final CDS. The first enhancement concerns Rule  $a$ , which now states that a node  $u$  decides to stay in  $C$  if it has the smallest ID among all its neighbors *and* it has two unconnected neighbors. This extended Rule  $a$  and the original Rule  $b$  generate a CDS for all possible networks except those whose topology graph is a clique (a case, again, that can be easily dealt with separately). The second enhancement pertains to the formation of  $C_u$  which now first includes all  $u$ 's neighbors for which  $u$  is not the neighbor with the smallest ID (called *free neighbors*). If the selection of the free neighbors still leaves some of the nodes in the two-hop neighborhood uncovered, additional nodes are added to  $C_u$  according to the MPR greedy algorithm. We observe that MPR-E (the extended version of MPR as proposed by Wu) generates a smaller CDS, and hence a better backbone, than the original MPR without imposing any additional communication overhead.

We finally describe those protocols that show a lower degree of localization, namely DCA, LRG, and WAF and we introduce the extensions to DCA and LRG, DCA-S and LRG-S.

The basic approach to building a CDS followed by these three protocols is (a) compute a (hopefully) small dominating set, and (b) connect it up.

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. 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 computes a dominating

set that it is  $O(\log n)$  times the optimum value, and that this is the best possible approximation that can be obtained in polynomial time, unless  $P = NP$ .

A distributed implementation of this heuristic has been given in [43]. Note however that this implementation exhibits a very low degree of localization since it is essentially a straightforward distributed implementation of the above sequential process. A “more parallel,” hence much faster, implementation of the greedy strategy is a non-trivial task, and it is based in a fundamental way on the use of randomness. In this paper we implement the randomized greedy heuristic introduced in [39]: *Local Randomized Greedy*, or LRG. The algorithm proceeds in rounds. The computation and communication in each round is the following. (1) Each node computes the highest span in its two-hop neighborhood. A node span is defined as the number of its neighbors currently uncovered. This step requires a node to transmit two messages: One for communicating its span and the second for broadcasting the bigger span in its neighborhood. (2) A node  $u$  is a candidate to the CDS  $C$  if its span  $d(u)$  is  $\geq d(v)$ , for each  $v$  that is in  $u$ 's two-hop neighborhood. If  $u$  is a candidate, it broadcasts a message to all its neighbors. (3) Based on this last message, every uncovered node  $v$  computes its support, which is the number of candidates that cover  $v$  (it may include itself). This number is sent to all neighboring candidates with another message. (4) Finally, each candidate  $u$  decides whether to enter  $C$  or not with probability  $1/med(u)$ , where  $med(u)$  is the average support of all its uncovered neighbors. The decision is broadcasted to all its neighbors. A node which is still uncovered, or that has some neighbors still uncovered, starts the process again in the next round.

The authors prove that LRG produces a CDS whose size is within  $O(\log n)$  from the optimum in poly-logarithmic time with high probability. The time bound is tight, since a network is provided in the paper where LRG takes at least a poly-logarithmic number of rounds to terminate (with high probability). Not surprisingly, this network is organized in layers. This organization induces the chain of dependency that limits LRG's degree of localization. From an implementation perspective, the fact that LRG requires quite a number of messages per round leads to a high message and energy overhead even if the algorithm takes only few rounds to complete successfully.

A different approach to (a) is to compute a maximal independent set of the network nodes. An *independent* set is a collection of nodes such that no two of them are neighbors. If an independent set is maximal then it is also a dominating set. This is the approach taken by both 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.

In the DCA protocol [11] all nodes are initially UNDECIDED and have associated a weight. Nodes weights induce a total ordering of the nodes (ties are broken by using IDs). During the execution of the algorithm

each node will decide to be IN or to be OUT (of the independent set). A node decides when all its neighbors with larger weight have decided. When the time comes, a node decides to be OUT if one of its neighbors is IN. Otherwise, it decides to be IN. (This simple scheme can be somewhat optimized by letting nodes to be OUT as soon as a neighbor is IN). The IN nodes form a maximal independent set, i.e., the minimal dominating set required for clustering. The execution time depends on possible chains of dependency between the nodes, which is why DCA exhibits a degree of localization lower than that of WuLi and MPR. As mentioned earlier, the worst case arises when the network topology is a chain of nodes whose weight are sorted in decreasing order: The node with the smallest weight has to wait for all other nodes in the chain. In practice however, as verified in the experimental section, this worst case does not happen and dependency chains tend to be reasonably short.

Once the dominating set (or independent set) is computed the task is to connect it to form a backbone (step (b)). The approaches used by DCA and WAF are of opposite nature. Let  $G = (V, E)$  be our network and let  $D \subseteq V$  be the independent set computed in step (a) by DCA. 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 can be proven that  $H$  is connected (see [24] for the proof). 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. This rule is also used to connect the nodes in the dominating set generated by LRG into a CDS.

The second approach to connect  $D$  is much more economical in terms of size, but it is computationally expensive. This is the approach followed by WAF [58]. 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,  $C = D \cup (\bigcup_{uv \in T} P(uv))$ . This protocol is implemented by first electing a leader  $r$  among the nodes, which is going to be the root of  $T$ . Then a tree 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 the tree the node from which it received  $m$  first. Each node also computes its rank, defined as the pair (level, node ID). When such process is completed, the root enters the maximum independent set and starts a color-marking process of the nodes, which proceeds layer by layer, to construct the maximal independent set based on the tree. Nodes in the maximal independent

set are then connected through the dominating tree  $T$ .

Distributed leader election is *enormously* expensive in practice, and exhibits a very low degree of localization. Computing a spanning tree or electing a leader are highly sequential 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 isolated 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) that we describe here tries to find a compromise between these two extremes. The algorithm described in [27] finds this compromise by deviating 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 contained in it. 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  $C = D \cup (\bigcup_{uv \in S} P(uv))$ . This sparsification procedure can be implemented in a completely asynchronous fashion. The communication cost is proportional to the maximum length of the cycles to be broken, since each edge needs to know if it is the smallest ID edge in a small cycle. We termed the DCA protocol enhanced with this “sparsificator” the DCA-S protocol. More specifically, we call DCA-S( $i$ ) the protocol obtained by applying the sparsificator to the DCA backbone where  $i$  is the maximum length of the cycles to be broken. LRG-S and LRG-S( $i$ ) are obtained from LRG in a similar way.

## 4 Protocols Comparison

We performed a thorough simulation-based performance evaluation of the protocols described in Section 3. The selected groups of protocols have been implemented in the VINT project network simulator (ns2) [56] and their performance have been compared to analyze the costs associated to clustering and backbone formation, as well as to assess the properties of the resulting backbone. In particular, we have considered the following metrics (all averages):

1. **Protocol duration**, i.e., the time needed by each protocol to complete clusterhead selection and backbone formation.
2. **The overhead per node** (in bytes) associated to the protocol operations. (These are physical layer measurements, which account for collisions and for the corresponding automatic packet retransmissions at the MAC layer.)
3. **Energy consumption per node**. This metric is important for assessing the application of clustering and backbone formation to networks whose nodes are energy constrained. If the energy and the overhead cost of maintaining a backbone is high, then clustering and backbone re-organization may impose a non-negligible burden to the network. On the other hand, clustering re-organization in terms of clusterhead and gateway rotation is needed to prevent nodes with more demanding roles from depleting their energy, possibly impairing network operations.
4. **Backbone size**, i.e., the fraction of the network nodes that are in the backbone (percentage). A smaller backbone is desirable for minimizing the routing overhead. In sensor networks with nodes that can turn off their radio interface (energy saving sleep mode) the backbone size is also a measure of how many nodes need to stay awake for data transport. Usually, the nodes in the backbone stay up or have a higher duty cycle for guaranteeing routes to the sink, and hence the smaller the backbone the more nodes can be in energy conserving mode.
5. **Route length** over the subgraph  $G_b$  of the visibility graph induced by the backbone construction. (There are no links between ordinary nodes). This metric gives a measure of how well a routing protocol can perform over the backbone.
6. **Backbone robustness**, defined as the number of backbone nodes whose removal (because of failure or energy depletion) causes backbone disconnection or the uncovering of ordinary nodes. This is the metric that provides an indirect measure on how long the network will be operational before requiring a backbone recomputation.



## 4.1 Simulation environment

Our ns2 implementation is based on the CMU wireless extension, i.e., on the IEEE 802.11 MAC whose parameters we have modified to take sensor nodes characteristics into account.

The simulations refer to scenarios in which  $n$  static 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  $\leq 30\text{m}$ . We call *visibility graphs* the topologies generated by drawing 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 [36]. 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 awake/asleep schedule of the nodes that keeps them up 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 over 300 connected visibility graphs.

## 4.2 Experiments

According to the experiments that we performed, the rest of this section is organized into three major parts. We started by comparing the performance of the considered variants of the WuLi and MPR protocols. In particular, we have investigated:

- The general Rule  $k$  to derive a CDS as proposed in [25]. In particular we have analyzed the effects of the parameter  $k$ .
- The impact on the backbone metrics of the different rules for computing the nodal weight, namely, when the weights equal the node IDs, or when they are based on the nodal degree as suggested by Stojmenovic in [55].
- The pros and cons of the different variants of the WuLi protocol proposed in [63], [25] and [55], as well

as of the variants of the MPR protocol presented in [1] and improved in [60].

This first comparative performance evaluation has the threefold aim of identifying the best performing protocols in this class, of showing the rationale behind the different ideas they are based on and behind their different performance, and of assessing the advantages of using protocols with high localization degree.

The second batch of simulations concerns the performance of those protocols with a lower degree of localization such as DCA [11], LRG [39] and their sparsified variants described above. Differently from the WuLi and the MPR protocols, in this second set of solutions nodes are prevented to fully execute the protocol operations in parallel because of “dependency chains” among nodes: Before deciding its role a node must receive messages about the role of some of its neighbors which in turn could wait to know the role of some of their neighbors and so on. Hence, although relying on local information exchange (the two hop neighborhood), these protocols exhibit a higher complexity which naturally results in higher message exchange and hence in higher costs. Our experiments aim at discovering which of the two approaches would result in backbones with a smaller number of nodes and shorter routes and in better performance with respect to all the other relevant metrics. We also investigate the sparsified version of both DCA and LRG, namely, DCA-S and LRG-S, respectively. The aim in this case is assessing the effectiveness of the sparsification rule in producing slimmer connected backbones, and to determine the corresponding overhead in terms of energy, transmitted bytes and time. Finally, we study the performance trade-offs associated to different tunings of key parameters of the sparsification technique. As before, based on this comparison we identified the best performing algorithm in this second class of protocols according to the selected metrics.

The third and final set of simulations compares the elected “champion protocols” from the first two sets with the WAF backbone formation protocol [58]. WAF, as mentioned, shows a low degree of localization, given that before deciding whether they will be part of the backbone or not nodes must wait for the election of a leader and the following tree construction. As mentioned earlier, despite this longer set up time and a higher message exchange, WAF forms extremely slim, tree-like backbones, hence our interest in comparing this solution with the best ones from the other classes.

This final comparison allows us to clearly picture the impact of the different degrees of localization showed by the three different classes of protocols we considered. We observe that the higher the degree of localization the simpler the protocols, which leads to lower complexity (in term of messages/byte exchanged and time) without compromising the capability to generate slim backbones with acceptably short routes. Therefore, a high degree of localization results to be a key element in the design of effective protocols for ad hoc clustering and backbone formation.

#### 4.2.1 Protocols with high degree of localization: WuLi and MPR

Our first set of experiments aim at evaluating the general Rule  $k$  proposed by Dai and Wu in [25]. In particular, we have studied the impact of the parameter  $k$  on the performance of the WuLi protocol. Results are depicted in figures 1 to 3.

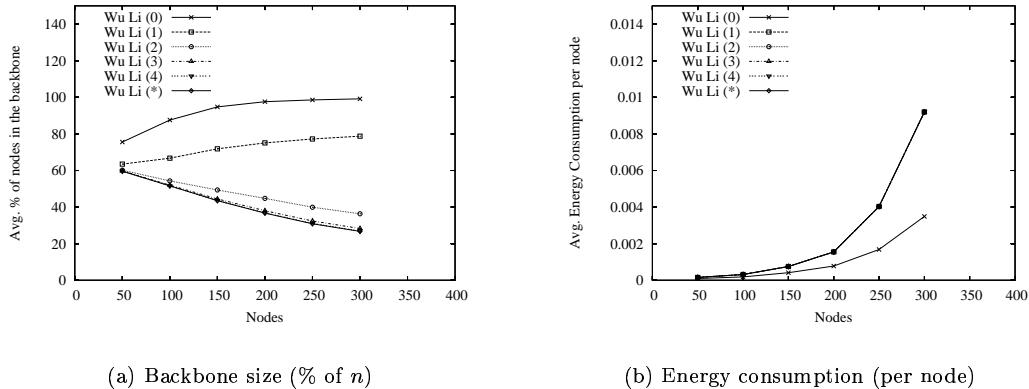


Figure 1: Average backbone size and energy consumption as functions of  $k$

Figure 1(a) depicts the backbone size for  $k = 0, 1, 2, 3, 4, \infty$  ( $k = \infty$  is indicated with a ‘\*’ in the figures). The case  $k = 0$  corresponds to the construction of the first, dense CDS: All nodes having two neighbors which are not in each other transmission range are in the CDS. As the figure clearly shows the application of the generalized rule with  $k = 0$  is not very effective. As  $n$  (and hence the network density) increases, it becomes more and more difficult that each neighbor  $v$  of a generic node  $u$  is directly connected to all of  $u$ ’s neighbors. This motivates the increasing backbone size when  $n$  increases. When  $n \geq 200$  basically all nodes are part of the CDS. The more  $k$  increases, the more effective the pruning rule: For removing  $u$  from the CDS  $C$  a group of up to  $k$  of its neighbors in  $C$  is required to cover  $u$  and its neighbors. In dense networks ( $n = 300$ ) the backbone obtained with  $k \geq 3$  is 64% smaller than the backbone obtained when  $k = 1$  and 71% smaller than the one obtained with  $k = 0$ .

Figures 1(b), 2(a) and 2(b) refer to energy consumption, bytes overhead, and time for backbone formation, respectively, while  $k$  is varying. With increasing network density the protocol performance degrades due to the higher number of nodes involved in the local exchanges of information, to the larger size of the message containing the neighbors list, and to the increased number of nodes competing for the radio channel. For all the considered scenarios, and independently of  $k$ , our results show excellent performance: WuLi is fast, has low overhead, and consumes little energy. This corresponds to the low complexity of the protocol, to the little amount of information that each node needs to exchange with its neighbors, and to the high degree of localization of this protocol. We observe that with respect to all these metrics, there is no significant

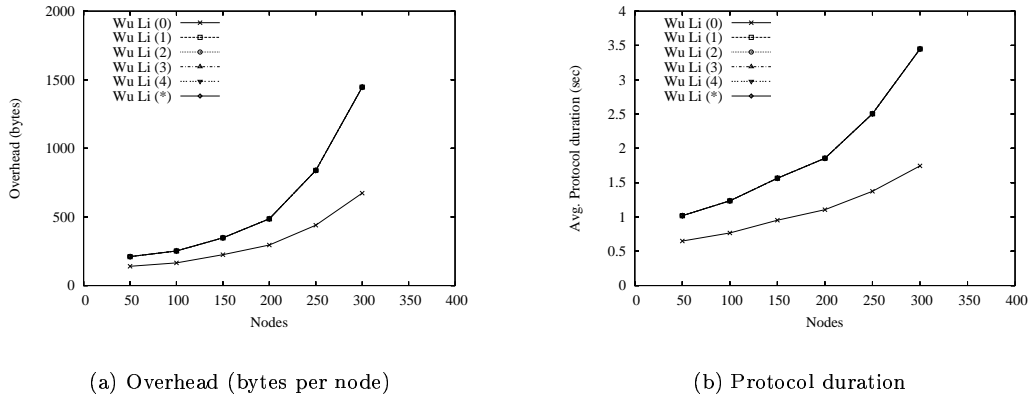


Figure 2: Average protocol overhead and duration as functions of  $k$

performance difference for  $k \geq 1$ . Once information about the two hop neighborhood is exchanged, each node decides (and communicates to its neighbors) whether it belongs to the connected dominating set or not. This, together with the knowledge of the two-hop neighborhood, is the sole information needed to apply the generalized rule. A higher  $k$  only implies a (limited) increase in computational complexity. However, no extra message exchange is needed. The case  $k = 0$  leads to reduced message overhead, shorter set up time and lower energy consumption since the pruning phase of the protocol is skipped. However, the produced backbone is remarkably denser, which detrimentally affects the network operations and its lifetime because of increased routing overhead and of the larger number of nodes that must be kept awake for guaranteeing a connected communication structure.

The average shortest paths length on the topology  $G_b$  induced by the backbone is depicted in Figure 3(a).

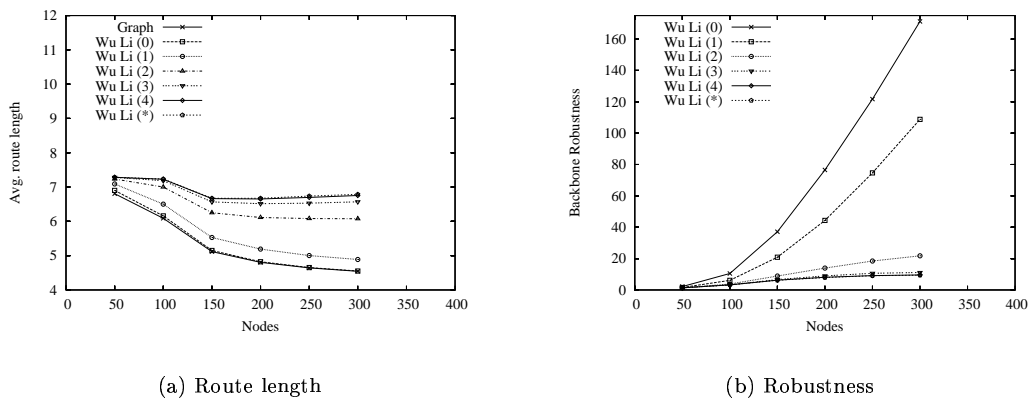


Figure 3: Average backbone route length and robustness as functions of  $k$

As  $k$  increases the average shortest path length also increases. We observed that the routes on  $G_b$  are up

$n$	50	100	150	200	250	300
WuLi(1)	2.48	3.89	6.03	8.29	10.76	13.2
WuLi(2)	2.28	2.79	3.26	3.53	3.63	3.71
WuLi( $\infty$ )	2.24	2.56	2.7	2.7	2.63	2.58
WuLi-D(1)	2.26	3.2	4.8	6.67	8.72	10.87
WuLi-D( $\infty$ )	2.17	2.55	2.86	3.01	3.08	3.18

Table 1: Average nodal degree on the backbone

to 48.6% longer than those of the visibility graph ( $n = 300$  and  $k \geq 4$ ). This reflects the fact the backbone topology gets sparser. Table 1 displays the average nodal degree in  $G_b$ . As expected, the higher the density the shorter the routes.

The higher or the lower density of  $G_b$  results in a higher or a lower robustness in case of backbone nodes failures. This is shown in Figure 3(b). When  $k \geq 3$  up to 10 randomly selected backbone nodes may die without affecting the backbone connectivity and the dominating property of the backbone. This number increases to over 20 when  $k = 2$ , and to over 100 in networks with 300 nodes when  $k = 1$ . Increased route length and decreased robustness are the price to pay for sparser backbones, leading to interesting, application-dependent trade-offs between these two different metrics.

We proceed in investigating the WuLi protocol with respect to different rules for weight selection, namely, whether we choose a backbone node based on its ID [25] or on its degree, à la Stojmenovic [55]. In what follows, WuLi-D( $k$ ) indicates the WuLi protocol when we use Rule  $k$  and nodes' weights are their degrees (the node IDs are used to break possible ties). WuLi( $k$ ) denotes the WuLi protocol when we use Rule  $k$  and nodes' weights are their unique ID. Results are depicted in Figure 4.

As expected, a degree-based weight leads to smaller backbones, as higher degree nodes are more likely to be able to cover a node and its neighbors, pruning it from the backbone (Figure 4(a)). While the WuLi( $k$ ) backbone contains the 51.6% of the  $n$  network nodes, the WuLi-D( $k$ ) contains only the 46% of them when  $n = 100$  and  $k \geq 3$ . We also have 36.7% (WuLi( $k$ )) vs. 33.6% (WuLi-D( $k$ )) when  $n = 200$  and  $k \geq 3$ , 66.7% vs. 55.6% ( $n = 100$  and  $k = 1$ ) and 75% vs. 63.2% ( $n = 200$  and  $k = 1$ ). This improvement is obtained at no extra cost in terms of overhead, energy consumption per node and protocol duration. The number of protocol messages exchanged is exactly the same independently of the choice of weight. (The degree of a node is obtained “for free” from the size of its neighbor list).

The degree-based weight selection results in denser backbones. This is confirmed by the values listed in Table 1, and it justifies why, for the same value of  $k$ ,  $k \geq 2$ , routes in  $G_b$  are shorter than routes obtained when nodes are selected to the backbone based on their ID (up to 8% shorter in case  $k \geq 4$ ).

Figure 4(c) displays the robustness of the backbone for WuLi-D( $i$ ). The comparison between these values

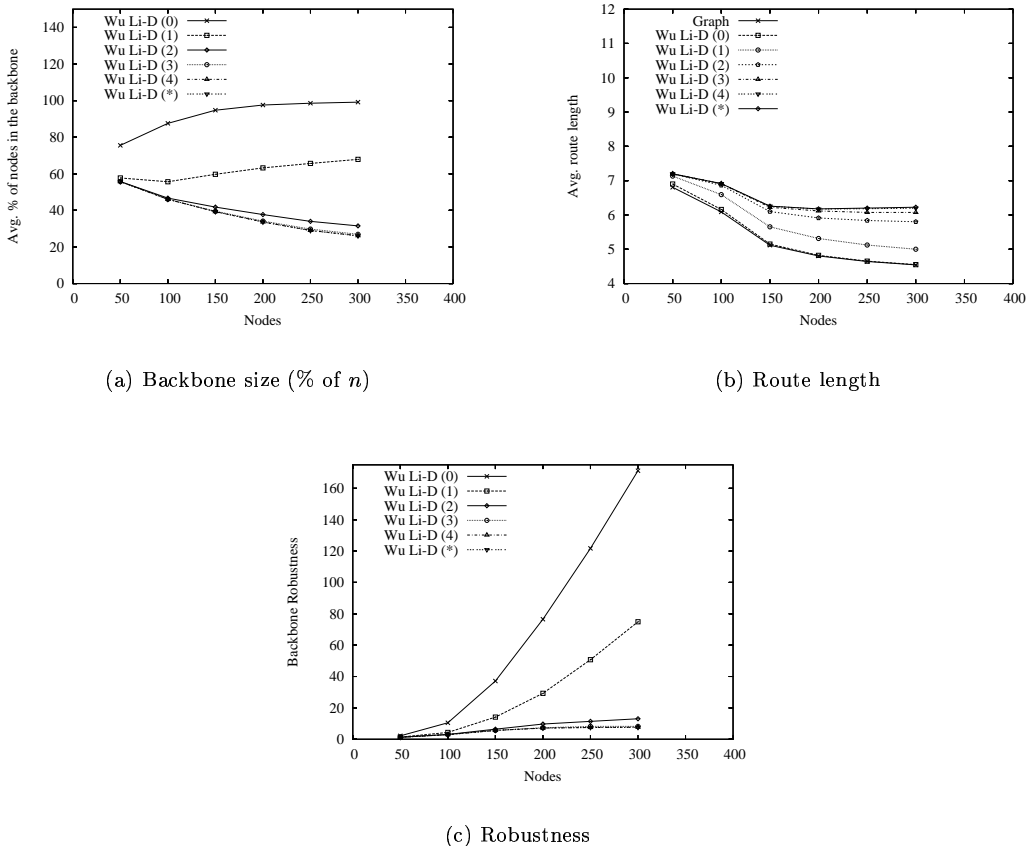


Figure 4: Average backbone size, route length and robustness for WuLi-D( $k$ ) as functions of  $k$

and the corresponding ones for WuLi( $i$ ) depicted in Figure 3(b) shows that, independently of  $k$  WuLi-D( $i$ ) is slightly less robust than WuLi( $i$ ). For higher  $k$ s (i.e., for denser backbones), this is explained by the fact that being the backbone nodes those with higher degrees, removing  $x$  of them corresponds to eliminating more links from  $G_b$  with respect to those removed from the WuLi( $i$ ) backbone.

The outcome of the previous experiments motivates the two new ideas proposed in [55] by Stojmenovic et al. Given that the first rule of the WuLi protocol does not bring to the construction of a small CDS, why do we have to compute such a big set? If we could just apply Rule  $k$  to the two hop neighborhood, we would save the message exchange that according to WuLi is needed for a node to communicate whether it is part of the CDS or not. This would have the beneficial effect of reducing protocol duration, protocol traffic and energy consumption. The second idea has to do with the backbone size: Why don't we choose the backbone nodes based on their degree? In the following we call "Stojmenovic" the protocol that combines these two ideas with WuLi's Rule  $k$  (where  $k$  has been set equal to  $\infty$ .)

In the last group of experiments for this first set of protocols we compare the best performing protocols

among WuLi( $i$ ) and WuLi-D( $i$ ) with Stojmenovic and with the extended version of MPR presented in [60], MPR-E. Results for the basic MPR protocols are not shown, since we observed that it is outperformed by MPR-E with respect to all the considered metrics. This is because MPR-E obtains a sparser  $G_b$  without affecting the protocol correctness and without requiring any extra message exchange. The main difference between the two versions is in the rule for the selection of the multipoint relays which favors the inclusion of nodes which are not likely to be included in the backbone later on. As a result, MPR and MPR-E have the same performance in terms of overhead, protocol duration and energy consumption. MPR-E, however, generates smaller backbones: The percentage of nodes in the backbone decreases from 52.47% to 48.22% when  $n = 100$ , and from 37.93% to 36.49% when  $n = 200$ . The price to pay is a very small loss in backbone robustness which decreases from 13 to 12 in dense networks ( $n = 300$ ).

Figure 5(a) depicts the backbone size of WuLi( $\infty$ ), WuLi-D( $\infty$ ), MPR-E and Stojmenovic for networks of increasing size ( $50 \leq n \leq 300$ ).

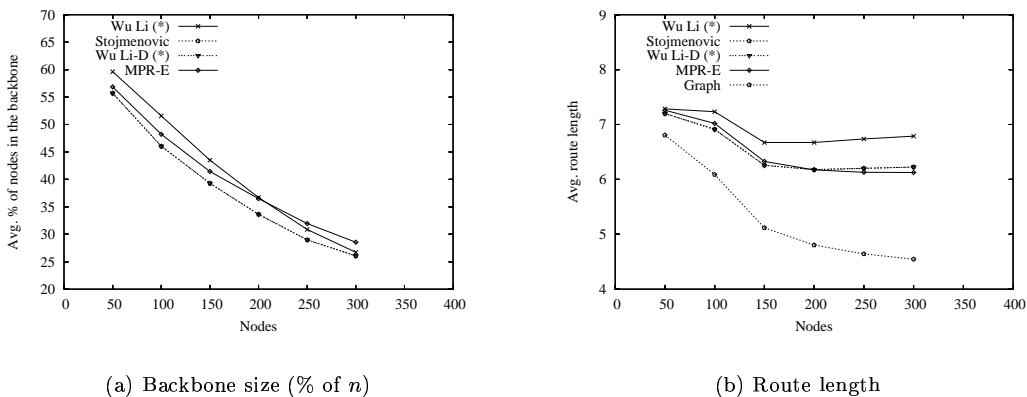


Figure 5: Average backbone size and route length on the WuLi( $\infty$ ), WuLi-D( $\infty$ ), Stojmenovic and MPR-E backbones

The WuLi-D( $\infty$ ) and Stojmenovic protocols have the best overall performance, which says about the clear advantage of a degree-based selection rule. We observe that the MPR-E protocol is not able to build a backbone as sparse as the one produced by WuLi-D( $\infty$ ) and Stojmenovic, especially when the network density increases: When  $n \geq 200$  MPR-E constructs backbones that are even bigger than those produced by WuLi (ID-based selection). In networks with 300 nodes the MPR-E backbone is, on average, around 10% bigger than the WuLi-D( $\infty$ ) and Stojmenovic backbones.

As clearly depicted in the figure, WuLi-D and Stojmenovic have exactly the same performance in terms of backbone size, independently of the value of  $k$  (here =  $\infty$ ). To explain this result requires us to investigate the details of the implementation of Stojmenovic. As mentioned, the idea is for a node to skip the phase

of communicating to its neighbors whether it belongs to the CDS or not. This, of course, does not mean that a node should not apply the first rule of the protocol, which allows it to check its membership to the CDS. If (and only if) a node  $u$  belongs to the backbone then it runs Rule  $k$  to see if it should be removed. The difference between WuLi-D( $\infty$ ) and Stojmenovic is in the set of neighbors which can cover node  $u$  and its neighbors. These are only the neighbors which said they are part of the CDS in WuLi-D( $\infty$ ), and *all* the neighbors in Stojmenovic. However, as proven by the following result, whenever a node  $u$  is removed because a group of neighbors with cardinality  $k$ , it exists a group of  $k$  connected neighbors of  $u$  that belong to the CDS, which cover  $u$  and  $u$ 's neighbors. In other words: *A node that is removed by the CDS by Stojmenovic is also removed by WuLi-D( $\infty$ )*. This proves the equivalence of the two protocols, since the “opposite” claim: *A node which is removed from the CDS by WuLi-D( $\infty$ ) is also removed by Stojmenovic* is straightforward (as the set to which Stojmenovic applies Rule  $k$  includes the corresponding WuLi-D set). In the following theorem we consider the CDS as formed by applying the first rule of WuLi, and we indicate with  $C(u)$  a minimal connected group of a node  $u$ 's neighbors which covers  $u$  and its neighbors.

**Theorem 1** *Let  $u$  be a node that runs Rule  $k$  according to the operations of Stojmenovic and is pruned. Then there exists a connected subset  $S$  of  $u$ 's neighbors such that 1)  $|C(u)| = |S|$ , 2) all the nodes in  $S$  are part of the CDS, and 3) the nodes in  $S$  cover  $u$  and  $u$ 's neighbors.*

**Proof** For sake of contradiction let us assume that there exists a node  $u$  such that all the minimal subsets  $S$  of cardinality  $|C(u)|$  contain at least a node  $x$  which is not part of the CDS. We distinguish two cases, depending on the cardinality of  $C(u)$ . Case 1:  $|C(u)| = 1$  ( $C(u)$  contains only  $x$ ). Since  $x \notin$  CDS, its neighbors are pairwise connected. Let  $N(x)$  denote  $x$ 's one-hop neighbors. As  $x$  covers  $u$  and its neighbors, these nodes belong to  $N(x)$ . Thus,  $u$ 's neighbors are pairwise connected. But this is impossible, since  $u$  applies Rule  $k$  (as all and only the nodes in the CDS after applying the first rule of WuLi). Case 2:  $|C(u)| \geq 2$ . In this case there must be at least a connected neighbor  $y$  of node  $x$  in  $C(u)$ . Since  $x$  is not in the CDS all its neighbors are pairwise connected. This implies that  $N(x) \subseteq N(y)$ . The set  $C(u) \setminus \{x\}$  still satisfies the rules for pruning node  $u$  (as it still covers  $u$  and its neighbors and it is still connected), and it is smaller than  $C(u)$ . This leads to a contradiction, since  $C(u)$  would not be minimal, as required. •

In terms of overhead, time and energy consumption MPR-E and Stojmenovic save over 50% with respect to the same metrics of WuLi( $\infty$ ) or WuLi-D( $\infty$ ). This is because Stojmenovic is able to skip the communications of the nodes in the CDS generated by the first WuLi rule. MPR-E, instead, obtains this improvement because of the very limited number of messages needed to build a local MPR set (a node  $u$  informs only its MPR neighbors for which it is the lowest ID neighbor of the fact they have been selected as MPR). MPR-E,



Stojmenovic and WuLi-D( $\infty$ ) also build backbones with shorter routes with respect to those of WuLi( $\infty$ ) backbones (up to 9% shorter).

In terms of backbone size the sparser the backbone the less robust it is. An average of 7.5 node failures are enough to disconnect the Stojmenovic backbone (or to isolate ordinary nodes) when  $n = 300$ . This number increases to 9.5 in the case of WuLi( $\infty$ ) and to 12 in MPR-E backbones.

As a representative of the protocols that exhibit a high degree of localization we choose Stojmenovic, given its slim backbones with short routes, its low overhead and energy consumption, and its fast operations.

#### 4.2.2 Lower degree of localization: DCA and LRG-based schemes

Figures from 6 to 9 display the performance, in terms of all the relevant metrics, of the DCA and LRG basic protocols as well as of their sparse versions DCA-S and LRG-S. With DCA-S( $i$ ) and LRG-S( $i$ ) we denote the protocols obtained by applying the sparsification rule proposed in [27] to the DCA and LRG backbones, respectively. We use the sparsification rule for breaking the cycles at most  $i$ -hop long in the auxiliary graph  $H(D)$ .<sup>2</sup> Such a cycle is broken by deleting its “smallest,” where smallest here is in accord to a total ordering on the set of the network links. Since discovering cycles is both energy consuming and time demanding we have limited our investigation to the case when  $i \leq 6$ . The cases when  $i = 4$  and  $i = 6$  are the only ones displayed in the figures since we observed that these two cases outperforms the cases when  $i = 3$  and  $i = 5$  in terms of all the relevant metrics. Consider DCA-S(3) and DCA-S(4) (the same reasoning applies also to LRG). These two protocols are equivalent in terms of energy consumption, the duration of the sparsification phase and byte overhead. This is because the discovery of  $i$ -hop long cycles,  $i = 3, 4$ , requires each clusterhead to send/receive messages to/from its neighbors clusterheads distant at most 2 hops in  $H(D)$ . Hence, the message complexity of the cases  $i = 3, 4$  and, similarly,  $i = 5, 6$  is the same. However, DCA-S(4) and DCA-S(6) lead to smaller backbones than DCA-S(3) and DCA-S(5), respectively. Hence, we only consider the former here.

Figure 6(a) compares the protocols performance with respect to producing a sparse connected backbone. All protocols obey the basic rule that the backbone size decreases as  $n$  increases: Growing network densities implies bigger cluster size and a wider choice of gateways. The DCA-S and LRG-S curves demonstrate the effectiveness of the sparsification rule applied to the DCA and LRG backbone. The percentage of nodes in DCA-S (LRG-S) backbones obtained by breaking triangles and squares ranges from 70% (65%) when  $n = 50$  to 22.6% (27.2%) when  $n = 300$ . With respect to the non-sparsified case, the DCA-S (LRG-S) backbones are 3% (1.3%) to 33.5% (24.8%) smaller. When cycles up to 6-hops long are broken, we obtain a further

---

<sup>2</sup> We recall that the graph  $H(D)$  has the cluster-heads as its nodes, and includes an edge  $(u, v)$  if and only if cluster-heads  $u$  and  $v$  are at most three-hops apart in the visibility graph.

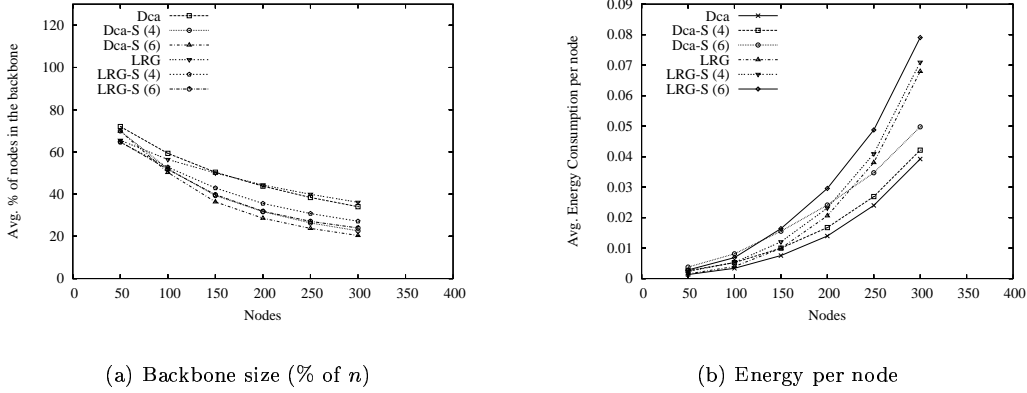
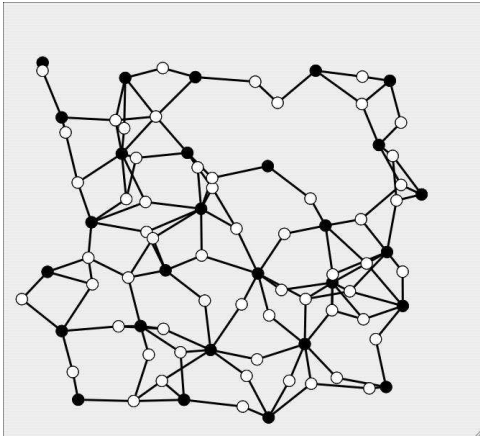


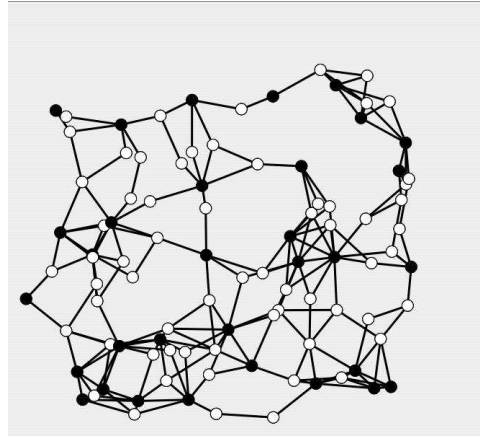
Figure 6: Average backbone size and energy consumed by DCA, LRG and their variants

10-11% reduction ( $n = 300$ ). In all considered scenarios, DCA and LRG have comparable backbone sizes and comparable number of clusterheads. The application of the sparsification rule however appears to be more effective on DCA-generated backbones. To justify this result we analyzed the number of *virtual links* with length one and two (crossing one or two gateways) in the backbones generated by DCA and LRG. (A *virtual link* is a path interconnecting adjacent clusterheads.) A gateway may have more than one virtual link incident to it. All these have to be eliminated for the gateway to be pruned by the sparsification rule. In Tables 2(a) and 2(b) we have listed the number  $Num_v(l)$  of virtual links of length  $l$  which were selected to be deleted by the sparsification rule. The number of virtual links for which the traversed gateways were actually removed from the backbone as a consequence of removing the virtual link is also listed (these tables are for the DCA-S(4) and LRG-S(4) protocols).

In case of virtual links of length two we distinguish between the number of virtual links in which only one of the two gateways was removed, denoted as  $Num_v(2, 1)$ , and the number of virtual links in which both the two gateways were removed, denoted as  $Num_v(2, 2)$ .  $Num_v(1, 1)$  indicates the number of virtual links traversing only one gateway that was removed. The number of virtual links of length one in the backbone generated by DCA is equal to 13 when  $n = 50$ , and 52.8 when  $n = 300$ . The number of virtual links traversing two gateways is 5.5 when  $n = 50$ . This increases to 25.8 when  $n = 300$ . LRG has a much smaller number of virtual links of length one (6.7 when  $n = 50$  and 44.5 when  $n = 300$ ), and a higher number of virtual links of length two (5.5 when  $n = 50$  and 33.5 when  $n = 300$ ). This reflects the fact that DCA, creating a set of independent nodes, distributes the clusterheads more evenly throughout the network, with a non-negligible number of clusterheads that are two hops apart (virtual links of length one). An example of this is shown in Figure 7 which compares the clusterheads (darker circles) in the DCA and LRG backbones that refer to the same sample topology (only clusterheads and gateways are shown in the picture).



(a) DCA clusterheads



(b) LRG clusterheads

Figure 7: Distribution of the clusterheads in the DCA and LRG backbones

In the dominating set produced by LRG clusterheads can be neighbors. We observe that LRG tends to produce groups of cluster-heads which are neighbors interconnected by three-hop paths. The values in Tables 2(a) and 2(b) confirm these observations.

$n$	50	300
$Num_v(1)$	0.55	19.55
$Num_v(1, 1)$	0.27	11.63
$Num_v(2)$	1.79	25.47
$Num_v(2, 1)$	0.62	11.31
$Num_v(2, 2)$	0.11	9.95

(a) DCA-S(4)

$n$	50	300
$Num_v(1)$	0.1	9.81
$Num_v(1, 1)$	0.04	3.7
$Num_v(2)$	0.78	29.41
$Num_v(2, 1)$	0.29	13.38
$Num_v(2, 2)$	0.05	7.88

(b) LRG-S(4)

Table 2: Virtual links removed by DCA-S(4) and LRG-S(4)

In particular, DCA-S(4) erases a higher number of virtual links of length one (from more than 6 times as many when  $n = 50$  down to three times as many when  $n = 300$ ) with respect to the number of virtual links of the same length deleted by LRG-S(4). The percentage of gateways belonging to a virtual link of length one selected for elimination which are actually removed from the backbone is also higher in DCA-S(4) than in LRG-S(4) (from two third down to one third when  $n = 300$ ). LRG-S(4) removes a higher number of virtual links of length two: An average of 29.41 at  $n = 300$  vs. the 25.47 removed on average by DCA-S(4). However, the advantage of removing more virtual links is balanced by the fact that DCA-S(4) successfully remove both gateways more often. The comparison between these results and the sizes of DCA-S(4) and LRG-S(4) backbones shows that the difference between backbone sizes is even more remarkable than the

difference between the number of virtual links whose gateways are removed. In the results displayed in tables 2(a) and 2(b) each gateway  $g$  pruned by the sparsification rule is counted multiple times: Once for each virtual link  $g$  belongs to. Thus, the fact that the difference in backbone size is higher than the difference between the number of deleted virtual links (the latter weighted according to the number of gateways actually removed) implies that, on average, gateways in LRG-S(4) belongs to a higher number of virtual links. This in turn justifies the fact that it is more difficult to actually remove nodes from the LRG backbone using the sparsification rule.

Figures 6(b), 8(a) and 8(b) depict the average energy consumption per node, the average protocol duration, and the overhead of DCA, LRG, DCA-S and LRG-S.

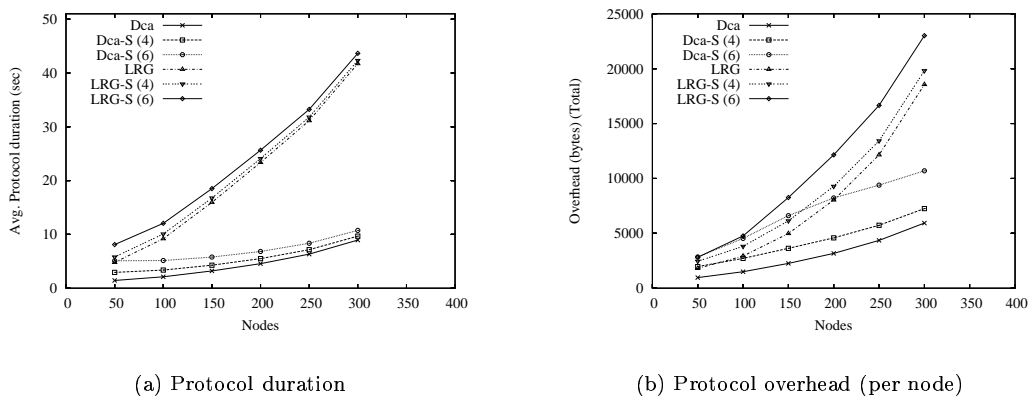


Figure 8: Average protocol duration and overhead in DCA, LRG and their variants

In terms of protocol duration the DCA (and DCA-S(4)) protocol(s) show reasonably good performance, requiring a time which ranges from 1.43s (2.93s) when  $n = 50$  to 8.94s (9.68s) 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 bigger weight to communicate their role before it can decide its own. Also, once clusterheads are elected, the messages exchanged between potential gateways and clusterheads for backbone construction may require significant information exchange, and hence extra time (although much less than for clusterhead selection). When  $n = 100$  ( $n = 200$ ) 1.2s (1.75s) are needed for joining clusterheads into a connected backbone.

It might appear counter-intuitive that DCA-S has a duration similar to that of DCA. This is because no dependency chains slow down the communication of information during the sparsification phase and that only a lower number of nodes participate in this phase. In other words, the sparsification phase has a high degree of localization. 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’s

name is balanced by many nodes quitting the protocol immediately after 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. Our experiments show that the S phase is quite short: No longer than 2.16s (average) in DCA-S(4). DCA-S(6) has worse performance than DCA-S(4) since the exchange of information to detect longer cycles require more time. In this case, the time needed by the backbone nodes for completing the S phase averages at 5.3s.

The duration of LRG is much higher than that of DCA. This is because LRG needs several rounds for clusterhead selection, and each round is implemented by the exchange of several messages (six messages are sent by every active node in a round). Since the number of active nodes decreases with the number of rounds, the time needed to complete each round becomes faster and faster. The average number of rounds needed by the LRG protocol ranges from 2 ( $n = 50$ ) to 3.5 ( $n = 300$ ). The corresponding protocol duration is up to 500% that of DCA. LRG-S does not take much longer than LRG for reasons similar to those discussed for DCA and DCA-S.

Figure 8(b) depicts the overhead of the different protocols for networks of increasing size. The advantages of a higher degree of localization is evident: DCA requires almost four times the bytes transmitted by WuLi. Due to its more involved operations LRG has a bigger overhead: Over three times as much as DCA's. DCA-S(4) and LRG-S(4) show a moderate increase in overhead with respect to DCA and LRG, respectively, since the number of nodes involved in the S phase of the protocol is quite limited. However, we observe a significant increase for DCA-S(6) and LRG-S(6), confirming the fact that identifying cycles is a time and message consuming operation: As the cycle length  $i$  increases the cost of identifying all the cycles up to  $i$  increases quickly. For the S phase to be practical the maximum length of the cycle has to be limited (we saw that  $i \leq 4$  is acceptable). Moreover, applying the sparsification rule over the DCA and LRG backbone is effective in decreasing the overhead of the S phase, maintaining the effectiveness of the sparsification rule.

As expected, energy consumption results follow closely those for the overhead, given the dependency of energy consumption on the number of bytes transmitted and received by each node. In networks with 300 nodes DCA-S(6), DCA-S(4), DCA, LRG-S(6), LRG-S(4), and LRG nodes require 0.049J, 0.042J, 0.039J, 0.079J, 0.07J and 0.067J, respectively.

Finally, Figure 9(a) and Figure 9(b) show the backbone robustness and the increase in the average route length with respect to the same metric in the visibility graphs. An increase in the average route 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 made clear in Figure 9(b) which shows an average increase in route lengths that ranges, in DCA

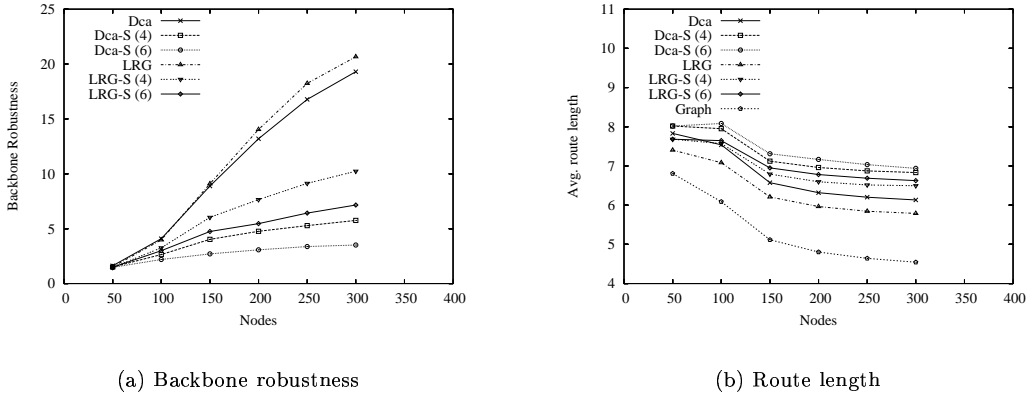


Figure 9: Average backbone robustness and increase in route length in DCA, LRG and their variants

backbones from 15% ( $n = 50$ ) to 36% ( $n = 300$ ), and, for LRG, from 8.8% ( $n = 50$ ) to 26.6% ( $n = 300$ ) with respect to the average route length in the visibility graph. DCA-S and LRG-S produce backbones that are sparser than DCA and LRG backbones, thus imposing routes which are even longer. DCA-S(4) has routes up to 11% longer than DCA's, while LRG-S(4) routes are an average 14% longer than LRG's.

As expected, the backbone robustness increases with the backbone density. In case of sparse visibility graphs, a single node failure may lead to disconnections even in the visibility graph. 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 re-routed to different backbone nodes. The denser the backbone, the more resilient it is to node failures. Dense and heavily meshed backbones such as DCA and LRG backbones may tolerate up to 20 failures in network with 300 nodes. This number reduces to only to 6 (10) for DCA-S(4) (LRG-S(4)).

Our second batch of experiment shows quite clearly that DCA and its S variant outperform the LRG protocols with respect to all selected metrics. The DCA-S results to be an effective way to further reduce the backbone size at the cost of additional time, energy and overhead. We observe that a good compromise between the extra cost and good performance is obtained when the length  $i$  of the cycle to be broken is 4. The backbone produced by DCA-S(4) is quite small and maintains an acceptable robustness at the price of a limited increase in energy consumption, overhead, time, and route length. Because of this we selected DCA-S(4) as the “champion” for this set of protocols.

#### 4.2.3 Comparison among protocols with different degrees of localization

In the third group of our experiments we compare the performance of the champions of the two sets of protocols previously investigated with the WAF protocol presented in [58].

This allowed us to show the effects of the different degrees of localization on protocols performance, and

to identify the best performing protocols among all the ones we considered.

Since the Stojmenovic idea of using a degree-based weight turned out to be effective in obtaining sparser backbones, for fairness of comparison we have implemented a variant of the DCA-S(4) where clusterheads are selected based on their nodal degree. We call this variant of the DCA the DCA-D-S(4) protocol.

The results of our comparison are depicted in figures 10 and 11. The average percentage of nodes in the backbones is shown in Figure 10(a).

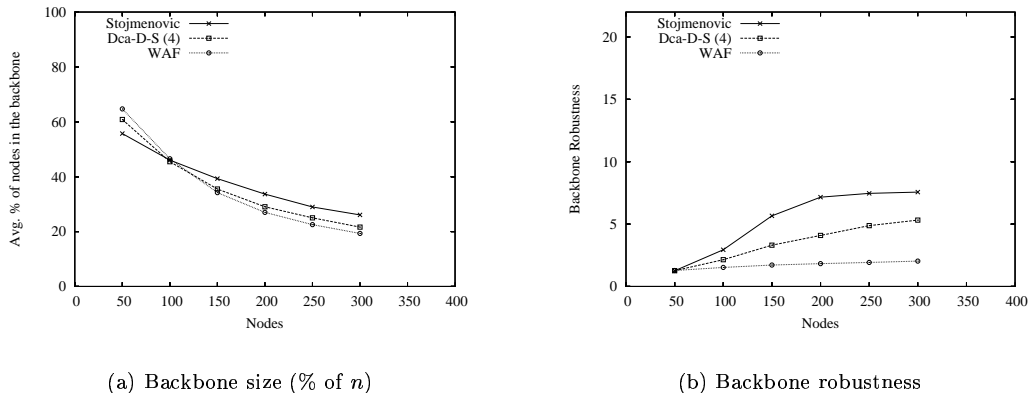
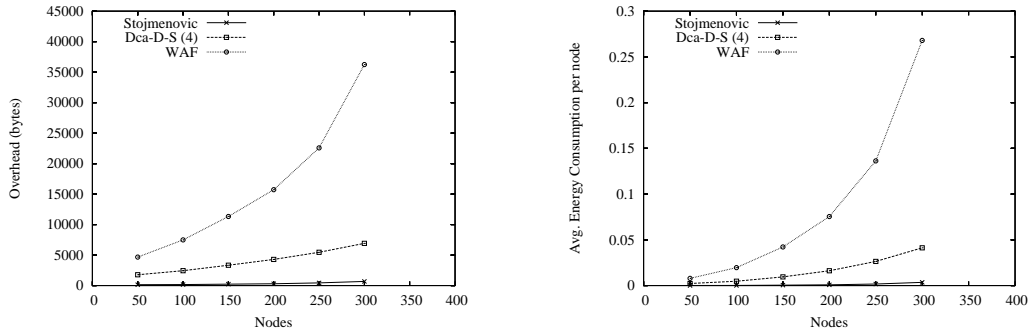


Figure 10: Average backbone size and robustness for DCA-D-S(4), Stojmenovic and WAF

WAF generates a thin “tree-like” backbone resulting, as expected, in a backbone considerably smaller than DCA-D-S(4) and Stojmenovic. However, the DCA-D-S(4) backbone size is only 11.8% bigger than WAF’s in the denser networks we considered ( $n = 300$ ). This clearly shows the effectiveness of using the described sparsification algorithm. Breaking cycles appears to be more effective than the sparsification rule used by Stojmenovic. The Stojmenovic backbones are up to 21% bigger, on average, than DCA-D-S(4) backbones. Slimmer backbones result in a slight increase in route lengths and in reduced robustness. The slim WAF backbone cannot survive many failures: The failure of two nodes is on average more than enough to disconnect it or leave some ordinary nodes uncovered. Although DCA-S generates backbones almost as thin as WAF backbones, its resilience to nodes failures is higher: An average of 6 failures can be tolerated without disconnecting the network ( $n = 300$ ). The denser Stojmenovic backbone survives up to 7.5 backbone nodes removals when  $n = 300$ .

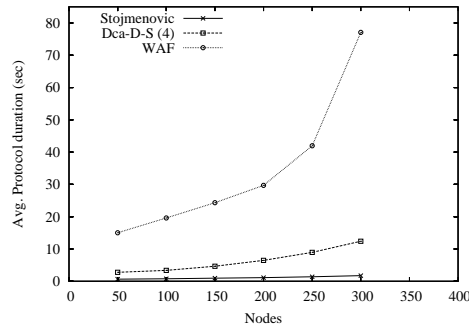
Figures 11(a), 11(b) and 11(c) show the overhead, energy consumption and time needed by the three protocols. These figures clearly show the gains that can be achieved by means of higher degrees of localization, and simple, “light” protocols.

Stojmenovic has a strong advantage with respect to these three metrics over all the other protocols. This is due to its low complexity, the reduced number of information that each node needs to exchange with its



(a) Protocol overhead (bytes sent per node)

(b) Energy consumption (per node)



(c) Protocol Duration

Figure 11: Average overhead, energy consumption and duration of DCA-D-S(4), Stojmenovic and WAF

neighbors, and the possibility to exchange this information in parallel (high degree of localization). The case of the WAF protocol clearly exemplifies the limits of an approach with very low degree of localization. It is by far the most energy consuming protocol, and requiring up to 77s to produce a connected backbone it is by far the slowest one. (Stojmenovic never requires more than 2s to construct the backbone.) The long duration of WAF is mostly 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 for progressively joining network fragments and to identify a common leader. The distributed implementation of this phase is time and overhead demanding, intrinsically sequential (very low degree of localization), and constitutes the real obstacle to adopt WAF in practical scenarios. In networks with 300 nodes it takes 67.5s to complete leader election. The 85% of the overall energy consumed is spent in this phase. In terms of message overhead, WAF requires up to 53 times the bytes of Stojmenovic. This is again due to the overly high complexity of the WAF leader election phase which requires information to be propagated to all the nodes in a fragment and to nodes in adjacent fragments every time two fragments are merged into a new one. When  $n = 300$  the leader election phase



WAF protocol	
h	Average backbone size
1	18.97
2	32.69
3	43.16
4	51.27

Table 3: Average backbone size (% of  $n$ ) as a function of the number  $h$  of sinks

accounts for 84.8% of the total WAF overhead.

One may observe that, in a wireless sensor network, the time and energy consuming leader election phase could be skipped by automatically designating the network sink as the leader. This significantly reduces overhead, energy consumption, and WAF duration. We observe that if the leader election phase can be avoided the WAF duration, its energy consumption and overhead are comparable to those of DCA-D-S(4).

Skipping the leader election phase is quite natural for WSN with a single sink. In case of multiple sinks operating independently, no changes in the protocols performance occur in DCA-D-S(4) and Stojmenovic. All sinks are connected to (included in) the produced backbone. In case of multiple sinks, the WAF protocol without the leader election phase has each sink constructing separate trees. The backbone obtained by merging the trees has potentially a very big number of nodes. Table 3 reports the average backbone size in case the backbone is obtained by interconnecting the dominating sets resulting from the execution of the WAF protocol by  $h$  sinks (randomly selected among the nodes),  $h = 1, \dots, 4$ . The  $h$  sinks run the backbone formation phase of the WAF protocol independently. The final backbone contains the union of the backbone nodes of the tree-like connected dominating sets generated by each of the sinks. As  $h$  increases, the backbone size rapidly increases, affecting the effectiveness of the network operations. The same trends were obtained by performing several other experiments in topologies with 300 nodes.

An example of two CDSs generated by randomly selected sinks and their union is displayed in Figure 12.

Overall our results show that the ‘‘S’’ rule for sparsification is a very effective technique for pruning CDSs without compromising their connectivity and dominance. In terms of backbone size the performance of protocols that can apply such rule is superior to the performance of all the variants of the WuLi and MPR protocols. However, when the other relevant metrics are taken into account, a higher degree of localization really makes the difference, significantly reducing the protocol duration, the overhead and the energy consumed for the backbone set up. Stojmenovic is an excellent compromise with respect to all these metrics.

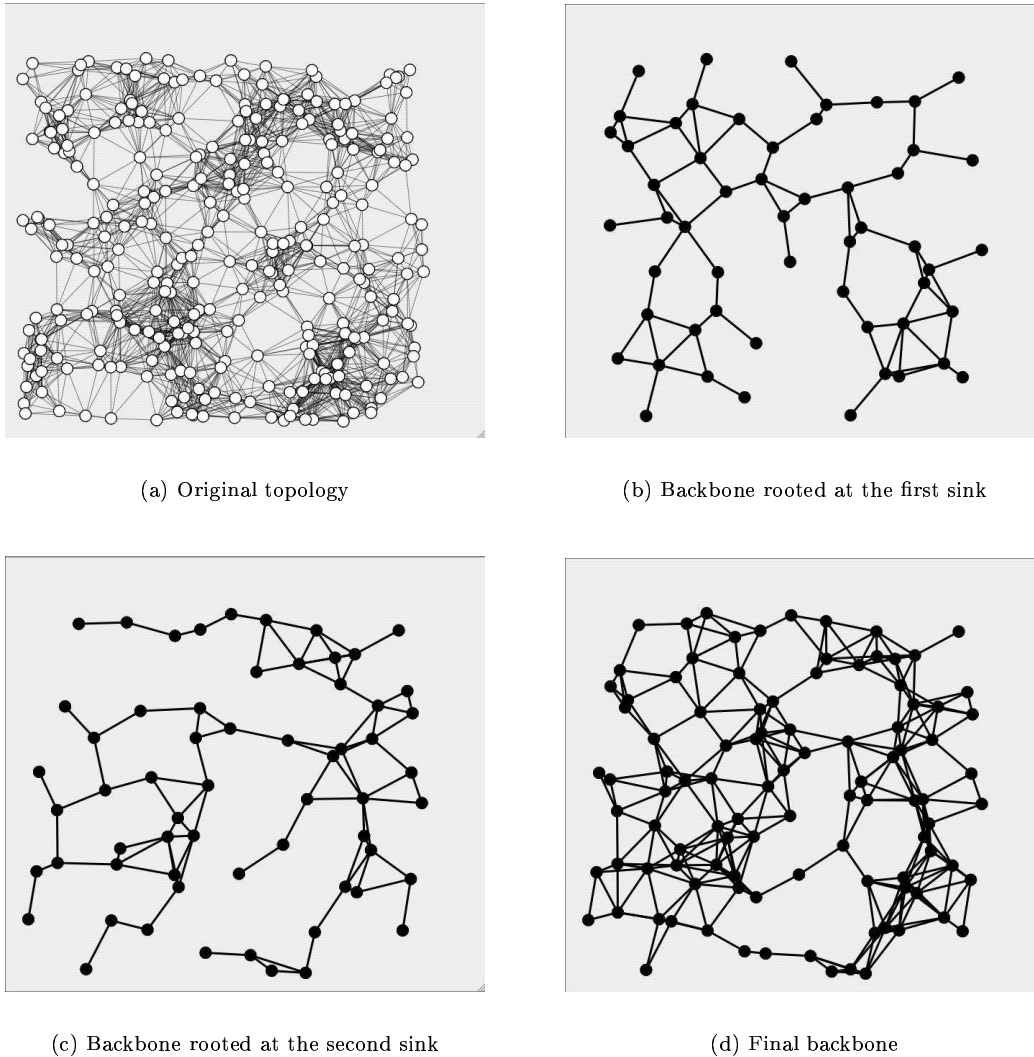


Figure 12: WAF backbones for a network with two sinks and their crowded union

## 5 Conclusions

In this paper we presented a thorough ns2-based comparative performance evaluation of protocols for clustering and backbone formation in large ad hoc networks with energy-constrained nodes. The selected protocols range from solutions that construct a rich connected backbone and then remove nodes that are unnecessary for maintaining connectivity and node coverage (such as the WuLi and MPR protocols) to techniques based on computing independent and dominating sets that are eventually connected into a backbone (e.g., DCA, WAF, and LRG).

We have classified the selected algorithms based on their “degree of localization,” which lead us to three different groups of protocols. The first group contains protocols that exhibit a high degree of localization

(WuLi and MPR). When nodes have gathered information about their (two-hop) neighbors they are immediately able to decide if they are going to be part of the backbone. In protocols like DCA and LRG (second group) nodes decide only when some of their neighbors have decided. This imposes a lower degree of localization since a node may wait for the decisions of nodes that are potentially far away from it. The final group of protocols that we have considered is represented by the WAF algorithm, which exhibits a very low degree of localization.

The experiments conducted have allowed us to thoroughly understand the behavior of each of the considered protocols, identifying the pros and cons of the different possibilities within each class. Based on the experiments we have chosen one “champion” protocol per class, i.e., the best performing protocol in terms of all the relevant metrics. A comparison between the performance of the chosen champions has then been performed, with the final aim to measure the performance gains corresponding to different degrees of localization.

Overall our results have shown that the “S” rule for sparsification is a very effective technique for pruning CDSs without compromising their connectivity. In terms of backbone size the performance of protocols that can apply such rule (e.g., DCA-S) is superior to the performance of all the variants of the WuLi and MPR protocols. However, when other relevant metrics are considered, a higher degree of localization is what makes the difference, significantly reducing the protocol duration, the overhead and the energy consumed for backbone set up. The Stojmenovic variant of the WuLi protocol is an excellent solution with respect to all these metrics.

## References

- [1] ADJIH, C., JACQUET, P., AND VIENNOT, L. Computing connected dominated sets with multipoint relays. *Ad Hoc & Sensor Wireless Networks* 1, 1 (2005). In press.
- [2] AKKAYA, K., AND YOUNIS, M. An energy-aware qos routing protocol for wireless sensor networks. In *Proceedings of the 23rd International Conference on Distributed Computing Systems Workshop, (ICDCSW 2003)* (Providence, RI, May 19–22 2003), pp. 710–715.
- [3] AKYILDIZ, I. F., SU, W., SANAKARASUBRAMANIAM, Y., AND CAYIRCI, E. Wireless sensor networks: A survey. *Computer Networks* 38, 4 (March 2002), 393–422.
- [4] AMIS, A. D., PRAKASH, R., VUONG, T. H. P., AND HUYNH, D. T. Max-min d-cluster formation in wireless ad hoc networks. In *Proceedings of IEEE Infocom 2000* (Tel Aviv, Israel, March 26–30 2000), vol. 1, pp. 32–41.
- [5] BAKER, D. J., EPHREMIDES, A., AND 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 (January 1984), 226–237.

- [6] BANDYOPADHYAY, S., AND COYLE, E. J. An energy efficient hierarchical clustering algorithm for wireless sensor networks. In *Proceedings of the 22nd IEEE Infocom 2003* (San Francisco, March 31–April 3 2003), vol. 3, pp. 1713–1723.
- [7] BANDYOPADHYAY, S., AND COYLE, E. J. Minimizing communication costs in hierarchically clustered networks of wireless sensors. In *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC 2003)* (New Orleans, LA, March 16–20 2003), vol. 2, pp. 1274–1279.
- [8] BANERJEE, S., AND KHULLER, S. A clustering scheme for hierarchical control in multi-hop wireless networks. In *Proceedings of the 20th IEEE Infocom 2001* (Anchorage, AK, April 22–26 2001), vol. 2, pp. 1028–1037.
- [9] BAO, L., AND GARCIA-LUNA-ACEVES, J. J. 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.
- [10] BASAGNI, S. Distributed and mobility-adaptive clustering for multimedia support in multi-hop wireless networks. In *Proceedings of the IEEE 50th International Vehicular Technology Conference, VTC 1999-Fall* (Amsterdam, The Netherlands, September 19–22 1999), vol. 2, pp. 889–893.
- [11] BASAGNI, S. Distributed clustering for ad hoc networks. In *Proceedings of the 1999 International Symposium on Parallel Architectures, Algorithms, and Networks (I-SPAN'99)* (Perth/Fremantle, Australia, June 23–25 1999), A. Y. Zomaya, D. F. Hsu, O. Ibarra, S. Origuchi, D. Nassimi, and M. Palis, Eds., IEEE Computer Society, pp. 310–315.
- [12] BASAGNI, S., CAROSI, A., AND PETRIOLI, C. Sensor-DMAC: Dynamic topology control for wireless sensor network. In *Proceedings of the 60th IEEE Vehicular Technology Conference, VTC 2004 Fall* (Los Angeles, CA, September 26–29 2004).
- [13] BASAGNI, S., ELIA, M., AND GHOSH, R. ViBES: Virtual backbone for energy saving in wireless sensor networks. In *Proceedings of the IEEE Military Communication Conference, MILCOM 2004* (Monterey, CA, October 31–November 3 2004).
- [14] BASAGNI, S., TURGUT, D., AND DAS, S. K. Mobility-adaptive protocols for managing large ad hoc networks. In *Proceedings of the IEEE International Conference on Communications, ICC 2001* (Helsinki, Finland, June 11–14 2001), vol. 5, pp. 1539–1543.
- [15] BELDING-ROYER, E. M. Multi-level hierarchies for scalable ad hoc routing. *ACM/Kluwer Wireless Networks* 9, 5 (September 2003), 461–478.
- [16] BETTSTETTER, C. The cluster density of a distributed clustering algorithm in ad hoc networks. In *Proceeding of the IEEE International Conference on Communications, ICC 2004* (Paris, France, June 20–24 2004), vol. 7, pp. 4336–4340.
- [17] BETTSTETTER, C., AND FRIEDRICH, B. Time and message complexity of the generalized distributed mobility adaptive clustering (GDMAC) algorithm in wireless multihop networks. In *Proceeding of the 57th IEEE Semiannual Vehicular Technology Conference, VTC 2003-Spring* (Jeju, Korea, April 22–25 2003), vol. 1, pp. 176–180.
- [18] BETTSTETTER, C., AND KRAUSER, R. 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.
- [19] BLUM, J., DING, M., THAELE, A., AND CHENG, X. Connected dominating sets in sensor networks and MANETs. In *Handbook of Combinatorial Optimization*, D.-Z. Du and P. M. Pardalos, Eds., vol. 1. Kluwer Academic Publisher, 2004, pp. 329–369.

- [20] CALLAWAY, JR., E. H. *Wireless Sensor Networks: Architectures and Protocols*. Auerbach Publications, Boca Raton, FL, August 2003.
- [21] CHAN, H., AND PERRIG, A. Ace: An emergent algorithm for highly uniform cluster formation. In *Proceedings of the 1st IEEE European Workshop on Wireless Sensor Networks, EWSN 2004* (Berlin, Germany, January 19–21 2004), H. Karl, A. Willig, and A. Wolisz, Eds., no. 2920 in LNCS, pp. 154–171.
- [22] CHATTERJEE, M., DAS, S. K., AND TURGUT, D. An on-demand weighted clustering algorithm (WCA) for ad hoc networks. In *Proceedings of IEEE Globecom 2000* (San Francisco, CA, November 27–December 1 2000), vol. 3, pp. 1697–1701.
- [23] CHEN, W.-P., HOU, J. C., AND SHA, L. Dynamic clustering for acoustic target tracking in wireless sensor networks. *IEEE Transactions on Mobile Computing* 3, 3 (July 2004), 258–271.
- [24] CHLAMTAC, I., AND FARAGÓ, A. A new approach to the design and analysis of peer-to-peer mobile networks. *Wireless Networks* 5, 3 (May 1999), 149–156.
- [25] DAI, F., AND WU, J. An extended localized algorithms for connected dominating set formation in ad hoc wireless networks. *IEEE Transactions on Parallel and Distributed Systems* 15, 10 (October 2004).
- [26] DAS, B., AND BHARGHAVAN, V. 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.
- [27] DUBHASHI, D., MEI, A., PANCONESI, A., RADHAKRISHNAN, J., AND SRINIVASAN, A. 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.
- [28] EPHREMIDES, A., WIESELTHIER, J. E., AND BAKER, D. J. A design concept for reliable mobile radio networks with frequency hopping signaling. *Proceedings of the IEEE* 75, 1 (January 1987), 56–73.
- [29] GARCIA NOCETTI, F., SOLANO GONZALES, J., AND STOJMENOVIC, I. Connectivity based  $k$ -hop clustering in wireless networks. *Telecommunication Systems* 22, 1–4 (2003), 205–220.
- [30] GERLA, M., AND LIN, C. R. 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.
- [31] GERLA, M., AND TSAI, J. T.-C. Multicluster, mobile, multimedia radio network. *Wireless Networks* 1, 3 (1995), 255–265.
- [32] GERLA, M., AND XU, K. Multimedia streaming in large-scale sensor networks with mobile swarms. *ACM SIGMOD Record* 32, 4 (December 2003), 72–76.
- [33] GUPTA, G., AND YOUNIS, M. Fault-tolerant clustering of wireless sensor networks. In *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC 2003)* (New Orleans, LA, March 16–20 2003), vol. 3, pp. 1579–1584.
- [34] GUPTA, G., AND YOUNIS, M. Load-balanced clustering of wireless sensor networks. In *Proceedings of the IEEE International Conference on Communications (ICC 2003)* (Anchorage, AK, May 11–15 2003), vol. 3, pp. 1848–1852.
- [35] GUPTA, G., AND YOUNIS, M. Performance evaluation of load-balanced clustering of wireless sensor networks. In *Proceedings of the 10th International Conference on Telecommunications (ICT 2003)* (Papeete, French Polynesia, February 23–March 1 2003), vol. 2, pp. 1577–1583.
- [36] HAVINGA, P. J. M., , ETALLE, S., KARL, H., PETRIOLI, C., ZORZI, M., KIP, H., AND LENTSCH, T. 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.

- [37] HEINZELMAN, W. R., CHANDRAKASAN, A., AND BALAKRISHNAN, H. 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.
- [38] HU, Y., HEIDEMANN, J., AND ESTRIN, D. 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.
- [39] JIA, L., RAJARAMAN, R., AND SUEL, T. An efficient distributed algorithm for constructing small dominating sets. *Distributed Computing* 15, 4 (December 2002), 193–205.
- [40] JOLLY, G., KUSCU, M. C., KOKATE, P., AND YOUNIS, M. A low-energy key management protocol for wireless sensor networks. In *Proceedings of the 8th IEEE International Symposium on Computers and Communications (ISCC 2003)* (Kemer-Antalya, Turkey, June 30–July 3 2003), pp. 335–340.
- [41] KOZAT, U. C., KONDYLLIS, G., RYU, B., AND MARINA, M. K. Virtual dynamic backbone for mobile ad hoc networks. In *Proceedings of the IEEE International Conference on Communications (ICC 2001)* (Helsinki, Finland, June 11–14 2001), vol. 1, pp. 250–255.
- [42] KRISHNA, P., VAIDYA, N. H., CHATTERJEE, M., AND PRADHAN, D. K. A cluster-based approach for routing in dynamic networks. *ACM SIGCOMM Computer Communication Review* 27, 2 (April 1997), 49–64.
- [43] LIANG, B., AND HAAS, Z. Virtual backbone generation and maintenance in ad hoc network mobility management. In *Proceedings of the 19th IEEE Infocom* (Tel Aviv, Israel, March 26–30 2000), vol. 3, pp. 1293–1302.
- [44] LIN, C. R., AND GERLA, M. Adaptive clustering for mobile wireless networks. *Journal on Selected Areas in Communications* 15, 7 (September 1997), 1265–1275.
- [45] MARCUCCI, A., NATI, M., PETRIOLI, C., AND VITALETTI, A. Directed diffusion light: Low overhead data dissemination in wireless sensor networks. In *Proceedings of IEEE VTC 2005 Spring* (Stockholm, Sweden, May 29–June 1 2005).
- [46] MCDONALD, A. B., AND ZNATI, T. 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* 17, 8 (August 1999), 1466–1487.
- [47] MHATRE, V., AND ROSENBERG, C. Design guidelines for wireless sensor networks: Communication, clustering and aggregation. *Ad Hoc Networks* 2, 1 (January 2004), 45–63.
- [48] MHATRE, V., AND ROSENBERG, C. Homogeneous vs. heterogeneous clustered sensor networks: A comparative study. In *Proceedings of the 2004 IEEE International Conference on Communications, ICC 2004* (Paris, France, June 20–24 2004), vol. 6, pp. 3646–3651.
- [49] NIEBERG, T., HAVINGA, P., AND HURINK, J. 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).
- [50] SANTI, P., AND SIMON, J. Silence is golden with high probability: Maintaining a connected backbone in wireless sensor networks. In *Proceedings of the 1st IEEE European Workshop on Wireless Sensor Networks, EWSN 2004* (Berlin, Germany, January 19–21 2004), H. Karl, A. Willig, and A. Wolisz, Eds., no. 2920 in LNCS, pp. 106–212.
- [51] SINHA, P., SIVAKUMAR, R., AND BHARGHAVAN, V. Enhancing ad hoc routing with dynamic virtual infrastructures. In *Proceedings of IEEE Infocom 2001* (Anchorage, AK, April 22–26 2001), vol. 3, pp. 1763–1762.

- [52] SIVAKUMAR, R., DAS, B., AND BHARGHAVAN, V. 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).
- [53] SIVAKUMAR, R., DAS, B., AND V., B. Spine-based routing in ad hoc networks. *ACM/Baltzer Cluster Computing Journal 1* (November 1998), 237–248. Special Issue on Mobile Computing.
- [54] SIVAKUMAR, R., SINHA, P., AND BHARGHAVAN, V. CEDAR: A core-extraction distributed ad hoc routing algorithm. *IEEE Journal on Selected Areas in Communications 17*, 8 (August 1999), 1454–1465.
- [55] STOJMENOVIC, I., SEDDIGH, M., AND ZUNIC, J. Dominating sets and neighbors elimination-based broadcasting algorithms in wireless networks. *IEEE Transactions on Parallel and Distributed Systems 13*, 1 (January 2002), 14–25.
- [56] THE VINT PROJECT. *The ns Manual*. <http://www.isi.edu/nsnam/ns/>, 2002.
- [57] TILLET, J., RAO, R., SAHIN, F., AND RAO, T. M. Cluster-head identification in ad hoc sensor networks using particle swarm optimization. In *Proceedings of the IEEE International Conference on Personal Wireless Communications, ICPWC 2002* (New Delhi, India, December 15–17 2002), pp. 201–205.
- [58] WAN, P.-J., ALZOUBI, K. M., AND FRIEDER, O. Distributed construction of connected dominating sets in wireless ad hoc networks. *ACM/Kluwer Mobile Networks and Applications, MONET 9*, 2 (April 2004), 141–149.
- [59] WANG, L., AND OLARIU, S. A unifying look at clustering in mobile ad hoc networks. *Wireless Communications and Mobile Computing 4*, 6 (September 2004), 623–637.
- [60] WU, J. An enhanced approach to determine a small forward node set based on multipoint relay. In *Proceedings of the 58th IEEE Semiannual Vehicular Technology Conference, VTC 2003-Fall* (Orlando, FL, October 6–9 2003), vol. 4, pp. 2774–2777.
- [61] WU, J., DAI, F., GAO, M., AND STOJMENOVIC, I. On calculating power-aware connected dominating sets for efficient routing in ad hoc wireless networks. *Journal of Communications and Networks 4*, 1 (March 2002), 1–12.
- [62] WU, J., AND LI, H. On calculating connected dominating sets for efficient routing in ad hoc wireless networks. In *Proceedings of the 3rd ACM International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (Dial M 1999)* (Seattle, WA, August 20 1999), pp. 7–17.
- [63] WU, J., AND LI, H. On calculating connected dominating sets for efficient routing in ad hoc wireless networks. *Telecommunication Systems, Special Issue on Mobile Computing and Wireless Networks 18*, 1/3 (September 2001), 13–36.
- [64] XU, K., HONG, X., AND GERLA, M. An ad hoc network with mobile backbones. In *Proceedings of the IEEE International Conference on Communications (ICC 2002)* (New York, NY, April 28–May 2 2002), vol. 5, pp. 3138–3143.
- [65] YOUNIS, M., AKKAYA, K., AND KUNJITHAPATHAM, A. Optimization of task allocation in cluster-based sensor networks. In *Proceedings of the 8th IEEE International Symposium on Computers and Communications (ISCC 2003)* (Kemer-Antalya, Turkey, June 30–July 3 2003), pp. 329–334.
- [66] YOUNIS, M., YOUSSEF, M., AND ARISHA, K. Energy-aware routing in cluster-based sensor networks. In *Proceedings of the 10th IEEE International Symposium on Modeling, Analysis and Simulations of Computer and Telecommunication Systems (MASCOTS 2002)* (Fort Worth, TX, October 11–16 2002), pp. 129–136.