Telecommun Syst
DOI 10.1007/s11235-007-9053-1

LIDAR: a protocol for stable and energy-efficient clustering of ad-hoc multihop networks

**Damianos Gavalas · Grammati Pantziou ·
Charalampos Konstantopoulos · Basilis Mamalis**

Abstract Clustering has been proposed as a promising method for simplifying the routing process in mobile ad hoc networks (MANETs). The main objective in clustering is to identify suitable node representatives, i.e. cluster heads (CHs) to store routing and topology information; CHs should be elected so as to maximize clusters stability, that is to prevent frequent cluster re-structuring. Since CHs are engaged on packet forwarding they are prone to rapidly drop their energy supplies, hence, another important objective of clustering is to prevent such node failures. Recently proposed clustering algorithms either suggest CH election based on node IDs (nodes with locally lowest ID value become CHs) or take into account additional metrics (such as energy and mobility) and optimize initial clustering. Yet, the former method is biased against nodes with low IDs (which are likely to serve as CHs for long periods and therefore run the risk of rapid battery exhaustion). Similarly, in the latter method, in many situations (e.g. in relatively static topologies) re-clustering procedure is hardly ever invoked; hence initially elected CHs soon suffer from energy drainage. Herein, we propose LIDAR, a novel clustering method which represents a major improvement over alternative clustering algorithms: node IDs are periodically re-assigned so that nodes with low mobility rate and high energy capacity are assigned low ID values and, therefore, are likely to serve as CHs. Therefore, LIDAR achieves stable cluster formations and balanced distribution of energy consumption over mobile nodes. Our protocol also greatly reduces control traffic volume of existing algorithms during clustering maintenance phase, while not risking the energy availability of CHs. Simulation results demonstrate the efficiency, scalability and stability of our protocol against alternative approaches.

Keywords Mobile ad hoc networks · Algorithm · Clustering · Stability · Energy efficiency

1 Introduction

Mobile wireless networking has enjoyed a dramatic increase in popularity over the last few years. The advances in hardware design, the rapid growth in the communications infrastructure, and the increased user requirement for mobility and geographic dispersion, continue to generate a tremendous need for dynamic ad hoc networking. Mobile ad hoc networks (MANETs) are an ideal technology to establish instant communication infrastructure for military and civilian applications in which both hosts and routers are mobile [21]. Such networks have dynamic, sometimes rapidly changing, random, multi-hop topologies. The goal of mobile ad hoc networking is to extend mobility into the realm of a set of wireless mobile nodes, where themselves form the network infrastructure in an ad hoc fashion [1].

D. Gavalas (✉)
Department of Cultural Technology and Communication,
University of the Aegean, Mytilene, Greece
e-mail: dgavalas@aegean.gr

G. Pantziou · B. Mamalis
Department of Informatics, Technological Education Institute of
Athens, Athens, Greece

G. Pantziou
e-mail: pantziou@teiath.gr

B. Mamalis
e-mail: vmamalis@teiath.gr

C. Konstantopoulos
Computer Technology Institute, Patras, Greece
e-mail: konstant@cti.gr

Dynamic routing is a key issue in MANETs design and deployment. However, it has been proved that a flat structure exclusively based on proactive or reactive routing schemes cannot perform well in a large dynamic MANET [14, 24]. In other words, a flat structure encounters scalability problems with increased network size, especially in the face of node mobility [16]. This is due to their intrinsic characteristics. The communication overhead of link-based proactive routing protocols is $O(n^2)$, where n is the total number of mobile terminals in a network [4]. This means that the routing overhead of such an algorithm increases with the square of the number of mobile nodes in a MANET. For a reactive routing scheme, the disturbing RREQ (route request) flooding over the whole network and the considerable route setup delay become intolerable in the presence of both a large number of nodes and mobility. Consequently, a hierarchical architecture is essential for achieving a basic performance guarantee in a large-scale MANET [21, 23].

However, it has been proved that a flat structure exclusively based on proactive or reactive routing schemes encounter scalability problems with increased network size, especially in the face of node mobility [16]. One promising approach is to build hierarchies among the nodes, such that the network topology can be abstracted. This process is commonly referred to as *clustering* and the substructures that are collapsed in higher levels are called *clusters* [6]. The notion of cluster organization is not new, in fact it has been used for multi-hop networks since their appearance [2, 9].

With the advent of MANETs the use of cluster architecture for multi-hop networks has been revisited. Clustering not only makes a large MANET appear smaller, but more importantly, it makes a highly dynamic topology to appear less dynamic [19]. Clustering is also crucial for controlling the spatial reuse of the shared channel (e.g. in terms of time division and frequency division schemes), for minimizing the amount of data to be exchanged in order to maintain routing and control information in a mobile environment, as well as for building and maintaining cluster-based virtual network architectures [6].

In clustering procedure, a representative of each cluster is ‘elected’ as a *cluster head* (CH) and a node which serves as intermediate for inter-cluster communication is called *gateway*. Remaining members are called *ordinary nodes* (see Fig. 1). CHs hold routing and topology information, relaxing ordinary mobile hosts (MHs) from such requirement; however, they represent network bottleneck points and—being engaged in packet forwarding activities—are prone to fast battery exhaustion. The boundaries of a cluster are defined by the transmission area of its CH.

A considerable body of literature has addressed research on MANETs clustering; many algorithms that consider different metrics and focus on diverse objectives have been proposed [3, 5, 11, 15, 17, 18]. Existing algorithms typically separate clustering into two phases, *cluster formation* and *cluster maintenance*, throughout the latter phase, initial cluster configurations may be modified, depending on nodes movement [16]. However, some clustering schemes employ explicit message exchange among MHs in *periodic* basis for maintaining the cluster structure [15, 17, 18]; that is, cluster formation is repeated at the end of each period resulting in excessive consumption of network resources. Yet, even the algorithms that apply a different cluster maintenance method may cause the cluster structure to be completely rebuilt over the whole network when some local events take place, e.g. the movement or “die” of a MH, resulting in some CH re-election (re-clustering) [3, 5]. This is called the *ripple effect* of re-clustering, which indicates that the re-election of one CH may affect the structure of many clusters and arouse the CH re-election over the network [7]. For clustering schemes with ripple effect, the communication complexity for the re-clustering in the cluster maintenance phase may be the same as that in the cluster formation phase and greatly affect the performance of upper-layer protocols.

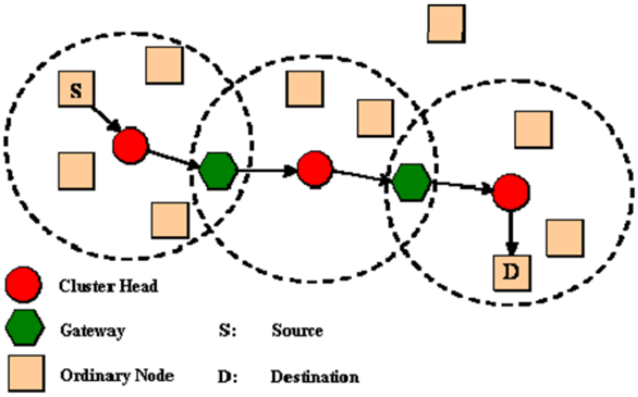


Fig. 1 Cluster heads, gateways and ordinary nodes in mobile ad hoc network clustering

In this article, we introduce a protocol for efficient and scalable clustering of MANETs designed with two main objectives in mind:

- Fast and inexpensive completion of clustering formation; our clustering algorithm incorporates both mobility and battery power metrics so that only MHs with low mobility and sufficient energy availability are likely to be elected as CHs; to meet this objective, we have extended a traditional clustering algorithm [17], described in the following section.
- Cost-effectiveness and ‘fairness’ in cluster maintenance; our algorithm aims at minimizing control traffic and enhance cluster stability, yet, not to prolong CHs serving time and cause rapid exhaustion of their energy supplies.

This article represents an extended version of the work presented in [12]. The remainder of the paper is organized as follows: Sect. 2 overviews related work and explains the motivation for our research. Section 3 describes the details of our proposed protocol, while Sect. 4 discusses simulation results. Finally, Sect. 5 concludes the paper and draws directions for future work.

2 Related work and motivation

Several heuristics have been proposed to address ad-hoc networks clustering problem. One of the most popular ones is the Lowest-ID (LID) [17], wherein each node is assigned a unique ID. Periodically, nodes broadcast their ID through a 'Hello' control message, within a period termed the 'Hello period' (HP). The lowest-ID node in a neighborhood is then elected as the CH; nodes which can 'hear' two or more CHs become gateways, while remaining MHs are considered as ordinary nodes.

Highest-Degree (HD) algorithm, originally proposed in [15], uses exclusively location information for cluster formation: the highest *degree* node in a neighborhood, i.e. the node with the largest number of neighbors, is elected as CH. Experiments have demonstrated that HD-based clustering suffers from poor cluster stability: the highest-degree node (the current CH) may fail to be re-elected even if it loses a single neighbor [5].

Vote-based clustering (VC) [18] uses both degree and power level information for CHs election, so as to prevent electing CHs with insufficient energy supply. However, simulation results reported in [18] revealed that the inclusion of the degree metric certainly affects clusters stability, similarly to the algorithm.

The main asset of LID method is its implementation simplicity. It is also a quick clustering method, as it only takes two HPs to decide upon cluster structure and also provides a more stable cluster formation than HD. In contrast, HD and VC need three HPs to establish a clustered architecture [18]. However, the main drawback of LID heuristic is its bias towards nodes with smaller IDs: these nodes are highly likely to serve as CHs for long periods which may lead to their rapid battery drainage. In addition, neither LID nor HD algorithm take into account mobility metrics, i.e. highly mobile nodes are equally likely to be elected as CHs, although their movement away from their attached cluster members may soon lead to a ripple re-clustering effect [23]. Most importantly, LID, HD and VC do not cater for separating cluster maintenance phase, i.e. CHs election takes place *periodically*; that scheme consumes considerable bandwidth so that upper-layer applications cannot be implemented due to the inadequacy of available resources.

The Weighted Clustering Algorithm (WCA) [5] employs combined-metrics-based clustering: a number of metrics, including node degree, CH serving time (to estimate residual energy capacity) and moving speed, are taken into account to calculate a weight factor I_v for every node v. Mobile nodes with local minimum I_v are elected as CHs. CHs election process is invoked: (a) at the very beginning of cluster formation; (b) during cluster maintenance, when a mobile node moves to a region not covered by any CH. WCA does not invoke re-clustering when a member node changes its attaching cluster. Even though this mechanism can enhance the stability of cluster topology, this also implies that CHs keep their status without considering the attribute of minimum I_v in later cluster maintenance. For instance, in relatively static networking environments, WCA will hardly ever be invoked, hence CHs service time will be prolonged and elected CHs will soon suffer from battery exhaustion. Also, the CH serving time alone is not a reliable indicator of energy consumption; hence, the accuracy of I_v values in WCA execution is in doubt [23]. Finally, article [5] does not clarify how MHs re-affiliation takes place, i.e. the process for the detachment of a MH from its current CH and the attachment to another [23].

In addition to the abovementioned methods, other approaches focus on cluster stability objective by identifying groups of MHs which exhibit similar mobility pattern (such groups of nodes are included into the same cluster, given that they are within transmission range of each other) [19, 22]. However, these works are not adequate for MANET environments where MHs show no group mobility behavior and are not relevant to our research.

A paper recently published by the authors of this article [11] introduced the Adaptive Broadcast Period (ABP) clustering algorithm, which mainly suits relatively static and quasi-static ad-hoc networks. The main objective of ABP is to relieve the network from the unnecessary burden of control messages broadcasting, especially in network topologies where MHs exhibit low mobility. ABP follows a completely different approach to that presented herein. In ABP, cluster election is based on the exchange of specially formatted 'Hello' messages and the calculation of MHs 'cluster head competence' value which considers location and energy availability metrics. Thereafter, the same CH election process is repeated in varying time intervals which depend on the mobility pattern of MHs; in fact, in highly mobile MANET environments, the performance of ABP, LID and HD has been shown to converge. Notably, no separation between cluster formation and cluster maintenance phases exists in ABP. The only similarity between ABP and the clustering protocol presented in this article lies on the method for estimating the MHs' mobility rate, described in Sect. 3.1.1. An extended version of [11] appears in [13].

3 Lowest-ID with adaptive ID reassignment (LIDAR) protocol

In this article, we propose a novel clustering protocol, Lowest-ID with Adaptive ID Reassignment (LIDAR). LI-

DAR explicitly separates cluster formation and cluster maintenance phases through employing two distinct algorithms. The former extends LID algorithm's approach to identify the most suitable CHs among MANET nodes in a fast and inexpensive manner. The latter aims at minimizing cluster re-formation occurrences, yet not at the expense of frequent network disconnections owned to CHs energy depletion. These two algorithms are presented in the following two sub-sections.

3.1 Cluster formation algorithm

The main idea behind LIDAR's cluster formation method is to maintain the assets of LID algorithm (fast, simple and low-cost clustering process) while providing stable clusters and catering for balanced computational load and power consumption among mobile nodes. This is achieved through identifying and electing the most suitable nodes as CHs, i.e. those with sufficient power level and low mobility rate.

MHs in a MANET normally depend on battery power supply, therefore energy consumption should be reduced in order to prolong the network lifespan [25]. Also, a CH bears extra work compared with ordinary members, and it is likely to "die" early because of excessive energy consumption. The lack of MHs due to energy depletion may cause network partition and communication interruption [6]. Hence, it is also important to balance the energy consumption among nodes to avoid node failures, especially when the network density is comparatively sparse.

In addition, mobility is a prominent characteristic of MANETs, and is the main factor affecting topology change and route invalidation [19, 22]. MHs that exhibit high mobility are inadequate for serving as CHs since their movement is likely to trigger frequent re-clustering, therefore increasing control traffic volume.

Therefore, our cluster formation algorithm takes into consideration both energy availability and mobility metrics to prolong network lifetime and avoid unnecessary re-clustering (i.e. enhance clusters stability). We have chosen not to include a node degree metric, as this has been shown to negatively affect cluster stability [11, 18, 23]. LIDAR's execution involves the following steps:

Step 1: At startup, node IDs are arbitrarily assigned. Initial clustering of mobile nodes is performed using LID algorithm, chosen due to its simplicity, fast and inexpensive completion of clustering process.

Step 2: At the end of every HP, each mobile node v calculates the following weighted function value:

$$W_v = N_1 w_1 B_v - N_2 w_2 M_{v,p}, \quad w_1 + w_2 = 1 \tag{1}$$

where B_v denotes the remaining battery life of node v and $M_{v,p}$ represents the mean mobility rate of node v during the latest p HPs, where p is a small integer (in the following sub-section, we describe how mobility rate is measured). Also, w_1 and w_2 are the weight coefficients of the energy and mobility metric respectively, while N_1 and N_2 are normalization factors.

Step 3: Whenever re-clustering is needed (in the following section we discuss the circumstances under which re-clustering process is triggered), CHs request their attached MHs to send their W_v values through a special broadcast message (WEIGHT_REQUEST).

Step 4: W_v values are unicasted by MHs to their local CH through a WEIGHT_REPLY message along with B_v values (the later are used during cluster maintenance phase).

Step 5: Having received W_v values from their attached cluster members, CHs sort them in descending order and re-assign node IDs so that small IDs are assigned to nodes with larger W_v values and large IDs to nodes with smaller W_v values. Namely, lower IDs are assigned to nodes with high power level and low mobility rate, thereby increasing their probability of being elected as CHs in the next algorithm's step.

Step 6: CHs send to their attached members their respective new_ID values through a CHANGE_ID message.

Step 7: Mobile nodes update their ID values. Right after, re-clustering procedure is invoked, where clusters formation is based on LID algorithm (go back to Step 1).

Certainly, it is likely that during the re-assignment of node IDs within individual clusters, it is likely that a MH first reports its W_v value to its current CH (Step 4), and then relocate into another cluster. Yet, even this scenario will not lead to a fatal error, as the CHANGE_ID message with its newly-assigned ID (Step 6) will be normally delivered by the routing protocol to that MH.

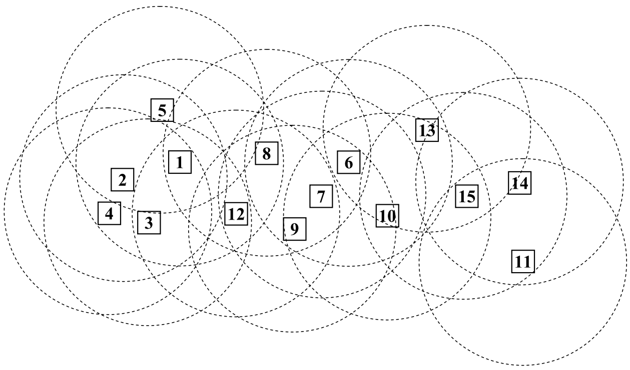
Upon completion of cluster formation, the protocol 'switches' to the cluster maintenance phase, i.e. control traffic is no longer exchanged until cluster formation process is re-invoked (details are given in the following section). LIDAR execution steps are illustrated in Fig. 2.

Table 1 presents how W_v values are calculated, where the coefficients of (1) are set to $w_1 = 0.7$ and $w_2 = 0.3$ (for this example we assume that $N_1 = N_2 = 1$). The preliminary ideas behind our cluster formation have been introduced in [10].

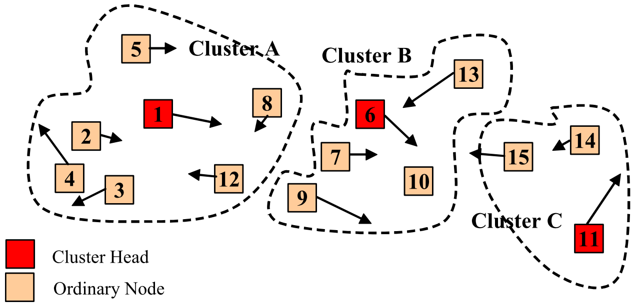
3.1.1 *Mobility rate measurement*

Most existing methods for estimating nodes mobility rate M pose the requirement for GPS card with sufficient accuracy mounted on every MH, e.g. [5]. We propose a novel, alternative method for measuring M which relaxes MHs from such requirement.

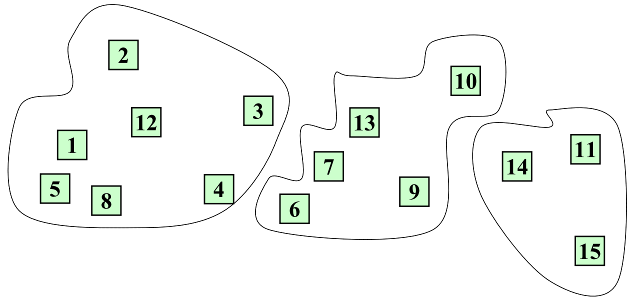
Each node v measures its own mobility rate M_v, used to calculate the weighted function value, in (1). This is



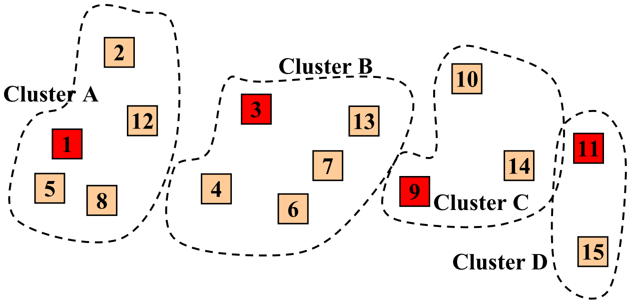
(**a**) Initial placement of mobile nodes on the plane (*dashed circles* indicate nodes transmission range)



(**b**) Initial clustering status of mobile nodes, based on Lowest ID (*arrows* depict velocity and direction of nodes movement)



(**c**) Re-assignment of node IDs within individual clusters



(**d**) Re-clustering of mobile nodes, based on Lowest ID

Fig. 2 Illustration of LIDAR execution steps

Table 1 Calculation of W_V values and node IDS re-assignment in LIDAR (where $W_1 = 0.7$ and $W_2 = 0.3$)

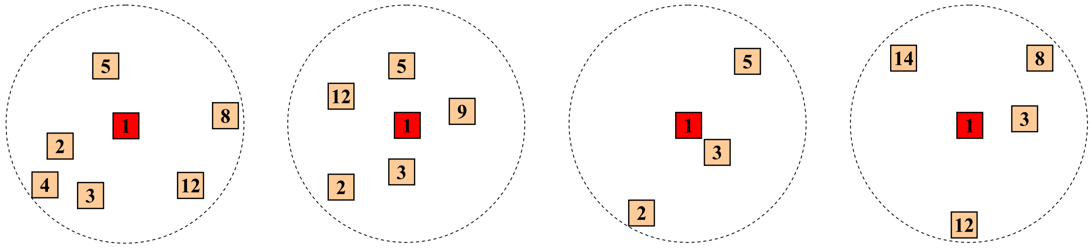
	Node ID	B_v	$M_{v,P}$	W_v	New Node ID
Cluster A	1	2	4	0.2	12
	2	7	1	4.6	1
	3	4	3	1.9	8
	4	6	4	3	5
	5	7	2	4.3	2
	8	6	1	3.9	3
	12	6	2	3.6	4
Cluster B	6	3	3	1.2	13
	7	7	2	4.3	7
	9	8	4	4.4	6
	10	6	0	4.2	9
	13	7	4	3.7	10
Cluster C	11	3	4	0.9	15
	14	6	1	3.9	11
	15	6	2	3.6	14

achieved through contrasting the topology information it obtains during successive HPs. Mobile nodes maintain a short ‘topology history table’ (THT); THT rows comprise vectors representing the IDs of neighboring nodes, where each THT row refers to different HP. Calculated M_v value actually represents the mean ‘vector distance’ among vectors recorded by v during the latest p HPs (where p is a small integer in order to minimize memory requirement): $M_{v,p} = (1/p)\sum_{i=0}^{p-1}(p-i)|\overline{\mathrm{THT}_{t-i\mathrm{HP}} - \mathrm{THT}_{t-(i+1)\mathrm{HP}}}|$, where t denotes the current time. The coefficient $(p-i)$ increases the weight of recent over older node movements on calculated $M_{v,p}$ values since the former are regarded as more reliable indicators of future mobility trends.

Figure 3 illustrates how mobile node with ID = 1 moves on the plane; as a result of that movement (and the movement of other network nodes), its neighboring nodes (i.e. those within its transmission range) differ at the end of every HP. For this particular example, the ‘neighborhood vectors’ of node #1 at the end of four successive HPs: are $\mathrm{THT}_1 = \{2, 3, 4, 5, 8, 12\}$, $\mathrm{THT}_2 = \{2, 3, 5, 9, 12\}$, $\mathrm{THT}_3 = \{2, 3, 5\}$, $\mathrm{THT}_4 = \{3, 8, 12, 14\}$. Hence, the mobility rate of node #1 within this period of time is given by:

$$\begin{aligned} M_1 &= (3 \times |\overline{\mathrm{THT}_4 - \mathrm{THT}_3}| + 2 \times |\overline{\mathrm{THT}_3 - \mathrm{THT}_2}| \\ &\quad + 1 \times |\overline{\mathrm{THT}_2 - \mathrm{THT}_1}|)/3 \\ &= (3 \times 5 + 2 \times 2 + 3)/3 = 7.33. \end{aligned}$$

Table 2 presents the THT of node #1, illustrating how its neighborhood changes and how its mobility rate is evaluated over those four successive BPs.



(a) (b) (c) (d)

Fig. 3 Neighboring nodes of node with ID = 1 during four successive HPs

Table 2 Instance of the topology history table of node #1 for the scenario illustrated in Fig. 3

HP	Neighboring nodes								MR
1	2	3	4	5	8		12		–
2	2	3		5		9	12		3
3	2	3		5					2
4		3			8		12	14	5

3.2 Cluster maintenance algorithm

The main criticism against cluster-based structures in MANETs focuses on the need for extra explicit message exchange among MHs for maintaining the cluster structure [16]. When network topology is highly dynamic, resulting in frequent cluster topology updates, the control overhead of cluster maintenance increases drastically. Thus, clustering operation may consume a large portion of network bandwidth, drain mobile nodes' energy quickly, and override its improvement on network scalability and performance [20]. By limiting re-clustering situations or minimizing explicit control messages for clustering, the cluster structure can be maintained well without excessive consumption of network resources [23].

Our cluster maintenance algorithm, follows an approach whereby clustering is not executed periodically but in an *event-driven* manner. That is, re-clustering process is only invoked when an important event occurs:

(a) The energy level of a CH has significantly decreased Each elected CH holds information about its node degree d and also the battery level B_v of its cluster members at the election time (see Step 4 of cluster formation algorithm). Nodes serving as CHs for a long period of time are expected to drop their battery level B_{CH} faster than ordinary nodes. To prevent the risk of energy depletion, CHs periodically check their B_{CH} value. When B_{CH} falls far below the average energy level of CH's cluster members, i.e. when

$$B_{\mathrm{CH}} < T \times \frac{\sum_{v=1}^{d} B_v}{d}$$

(where $T \leq 1$), the CH invokes a cluster formation process; namely, the CH is soon replaced by another node with higher energy availability. Unlike the method proposed in [5], our approach ensures that CH role is fairly shared among MHs regardless of the MANET's topology characteristics, hence energy consumption is uniformly distributed. It is stressed that our proposed scheme does not cause a ripple of re-clustering effect, since only CHs with decreased battery level relinquish their CH role, without affecting neighboring clusters. Admittedly though, this method may lead to portioning of clusters in which the CH gives up its role, thereby resulting on temporarily longer routing paths. In Fig. 2b for instance, assuming that node #1 stops serving as CH, if nodes #4 and #8 are not within communication range and node #4 is elected as the new CH, the cluster will be partitioned. However, MHs remote to the newly-elected CH may join neighboring clusters, given that they are within communication radius of the corresponding CHs (e.g. node #8 may become member of cluster B). Our simulations tests have shown that in the long term, the initial number of clusters is nor significantly increased over time and basically stabilizes after a small number of re-clustering process invocation.

An alternative approach for relating re-clustering process with the event of CHs energy drainage (we intend to evaluate this approach in the future) would be to invoke re-clustering when CHs have transmitted a certain volume of data; this idea is based on the reasonable assumption that the energy level of CHs is not reduced relatively to the energy of their assigned nodes, as soon as no data are transmitted or received. That way, cluster members would not be required to inform their CHs about their battery level.

(b) The MANET topology has significantly changed The highly dynamic nature of MANET topologies combined with infrequent re-clustering implies that cluster structures may soon be outdated. On the other hand, the maintenance of updated cluster formations presupposes frequent exchange of control traffic, which should certainly be avoided. Hence, we propose a scheme whereby cluster formation is invoked when the MANET topology has changed to such extent that CHs are unable to route incoming traffic to its

Table 3 Simulation parameters

Parameters	Value
Simulation time	300 sec
MAC protocol	IEEE 802.11
Simulated plane	1000 ×1000 m^2
Transmission range	71 m
Number of MHs	100
Average speed of MHs	5 m/sec
Average rate of transmission requests for MHs	1 request/min
Hello period (HP)	0,5 sec
Weight of battery metric (w_1) in the calculation of W_v	0,7
Weight of mobility metric (w_2) in the calculation of W_v	0,3
Initial battery lifetime	20–100 units
Threshold of CHs energy level (T) used to invoke re-clustering process	0,5
Period of mobility rate monitoring (p) by CHs	5 HPs

destination node. Following that approach, we ensure that in relatively static MANET topologies (e.g. in convention centers, conferences or electronic classrooms), where relocations of MHs seldom occur, the cost of cluster maintenance is practically eliminated. However, this enormous cost improvement is achieved at the expense of larger setup latency whenever data traffic exchange commences. An alternative method would be to invoke re-clustering whenever a MH re-affiliates (moves away from its attached CH and joins another cluster). Such a method though, would generate excessive control traffic exchange in highly mobile networks for cluster maintenance; in most cases, control traffic would be broadcasted for no reason, e.g. MHs continuously changing their location on the plane, yet, not transmitting any data.

To illustrate our method, let us examine the example topology of Fig. 2d, which depicts the result of executing our cluster formation algorithm. At a later stage we assume that node #12 issues a data transmission request. At that time, network topology is expected to have changed due to nodes mobility. If this is not the case (topology has remained unchanged), node #1 (nominated as CH of node #12 at cluster formation time) will receive the transmission request and reply sending back an ACK message. Node #12 will than commence data transmission and CH #1 will route received data towards its destination node. If the transmission request is not received by node #1 (either node #1 or #12 has moved away), node #12 will not receive back the ACK message; as soon as a specified period of time elapses, node #12 will have detected the topology change and trigger a local re-clustering process. The outcome of re-clustering will be the attachment of node #12 to another CH; data transmission will start thereafter. Re-clustering process is 'propagated' along data routing path, if needed. That implies that our approach prevents the ripple re-clustering effect, since re-clustering is only invoked where necessary, i.e. in MANET areas that appear to have significantly reformed.

4 Performance evaluation through simulations

LIDAR protocol has been simulated using NS-2 simulator[1] and compared against LID, HD and WCA algorithms. Our simulation tests attempt to compare the performance of these algorithms in terms of signaling traffic, cluster stability and variance of MHs energy level. The parameters used in our simulation tests are those shown in Table 3, unless otherwise specified.

A square terrain of 1000 m × 1000 m is assumed. The number of MHs moving within the square space varies from 20 to 120. At startup, MHs are randomly positioned on the plane. MHs move with speed 0–15 m/s, on random direction. At the event of reaching the terrain boundary, MHs are bounced back. The 'hello period' duration is set to 5 ms for LID, HD and cluster formation phase of WCA and LIDAR approaches. Initial remaining battery time of MHs is randomly set between 20 and 100 units; energy is assumed to be linearly decreased for ordinary nodes, while for CHs it depends on the number of their attached cluster members. Each simulation run lasts 3 minutes; simulation results presented below have been averaged over 5 runs. Regarding the execution parameters of LIDAR, W_v values are calculated for $w_1 = 0.7$ and $w_2 = 0.3$; MHs measure their mobility rate through contrasting the topology information they obtain during $p = 5$ successive 'hello periods' CHs check their battery availability B_{CH} with a period 100 times longer than the 'hello period'.

Figure 4 illustrates the average number of control messages exchanged as simulation time advances. In LID and HD algorithms, 'Hello' messages are periodically broadcasted during cluster maintenance phase; hence, their performance results coincide. WCA executes re-clustering whenever a MH moves to a region not covered by any CH [5]. On the other hand, the most likely scenario for LIDAR re-clustering is when a MH issues a transmission request. Thus, for reasonable values of average MHs speed (5 m/sec) and average rate of transmission requests (1 request per min for each MH), LIDAR clearly outperforms WCA.

Figure 5 reveals the dependency of WCA algorithm's performance on the average speed of MHs. Namely, in highly mobile MANET environments WCA involves frequent re-clustering, hence increasing clustering overhead.

---

[1] NS-2 (Network Simulator), http://www.isi.edu/nsnam/ns/.

Fig. 4 Average number of control messages during the simulation time

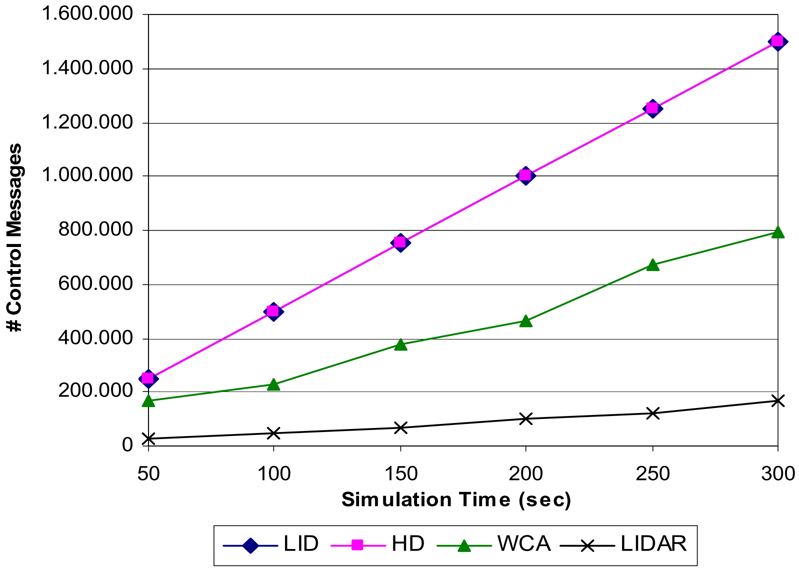
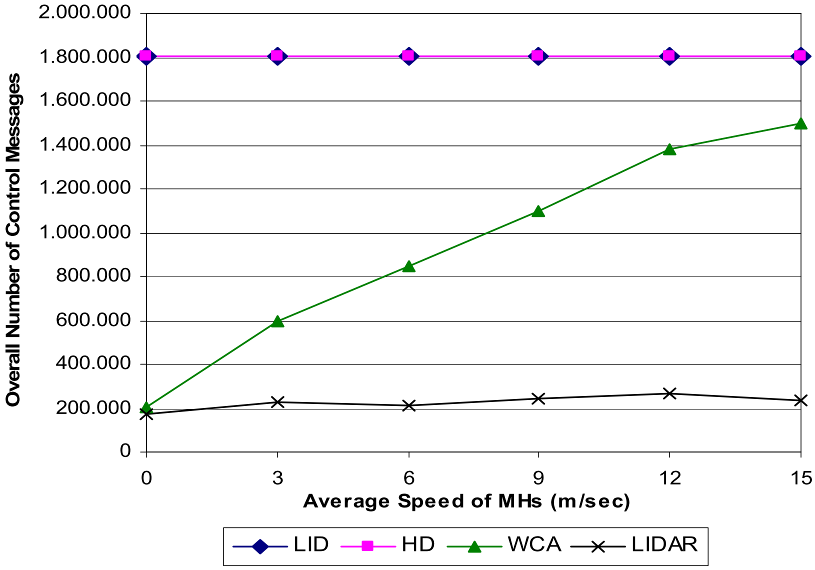


Fig. 5 Overall number of control messages as a function of MHs average speed



In contrast, LIDAR's performance remains unaffected; yet, it depends on the frequency of transmission requests. This is shown in Fig. 6, which depicts a linear increment of LIDAR's control overhead with transmission requests rate.

Figure 7 compares the average number of CH changes, which is an indicator of the overall cluster structure stability (the more frequent the CH changes, the less stable clusters are). As expected, LID performs better than HD as the former exclusively uses ID and the latter node degree information to decide upon cluster structure. WCA also incorporates degree metric in cluster formation thereby negatively affecting cluster stability; also, as network size increases, it is more likely to invoke re-clustering process due to nodes movement. LIDAR provides better results, as it suggests that CH changes do not depend on nodes mobility but may only occur upon data transmission or when CHs run the risk of battery drainage.

Figure 8 illustrates the variance of power level among MANET's MHs. Large variance values indicate that specific nodes are engaged on CH role for long periods, hence, their energy level soon falls far below the average. This simulation test highlights the main limitation of LID algorithm: in LID, CHs election is biased in favor of nodes with low ID values; these nodes are likely to serve as CHs for long time and their energy supply rapidly depletes. Interestingly though, for static environments (average speed 0 m/sec), LID, HD and WCA algorithms present almost identical variance values among MHs energy level. For LID and HD methods cluster formation is periodically executed only to re-elect the same nodes as CHs (since network topology

Fig. 6 Overall number of control messages as a function of MHs average rate of transmission requests

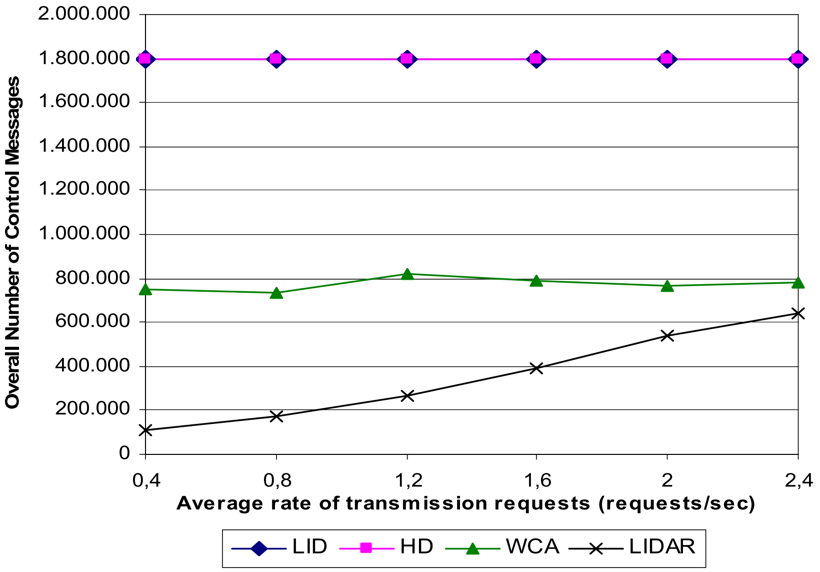
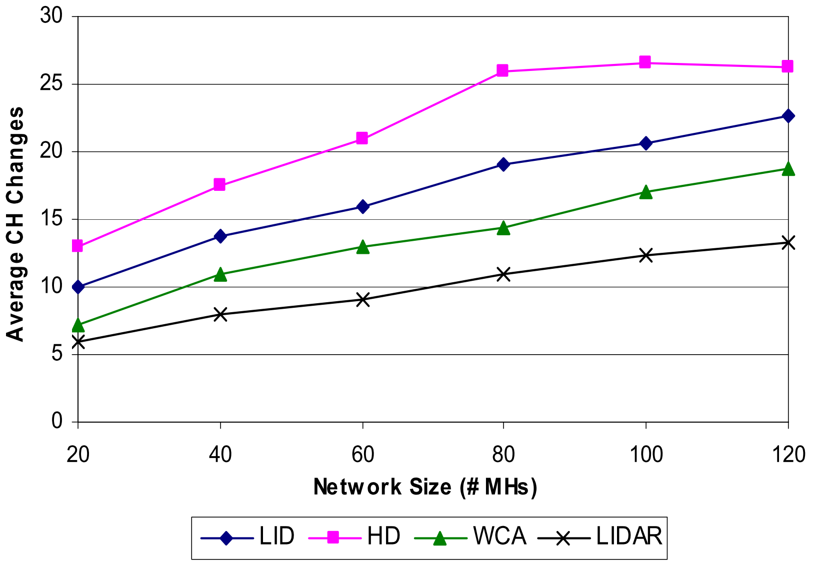


Fig. 7 Average number of CH changes as a function of network size



does not reform). For WCA, following the initial cluster formation, the lack of nodes movement prevents future re-clustering, hence CHs service time is prolonged and difference between the energy levels of CHs and ordinary nodes increases. However, higher mobility rates imply more frequent triggering of WCA re-clustering events, thereby decreasing variance values. LIDAR exhibits smaller variance of mobile nodes energy level: CHs give up their role even in static environments, when their battery resources are about to exhaust. Namely, CHs role is fairly shared among network nodes, achieving more uniform distribution of energy consumption.

Figure 9 draws the number of node failure occurrences throughout the duration of simulation runs. This parameter is crucial since a large number of node failures is likely to cause network partitioning and communication interruption [6]. LID presents an exponential increment of node failures; its performance is almost comparable to that of WCA for static MANET environments (in this scenario, WCA initial clustering remains unchanged, hence most of the CHs exhaust their battery power by the end of the simulation). Yet, WCA performance improves for higher mobility rates (5 m/sec). On the other hand, the number of node failures in LIDAR is practically unrelated to the mean speed of MHs; our algorithm outperforms all alternative approaches due to the invocation of re-clustering process by CHs that are about to fail because of the consumption of their energy supplies.

Fig. 8 Variance of energy level among MHs as a function of their average speed

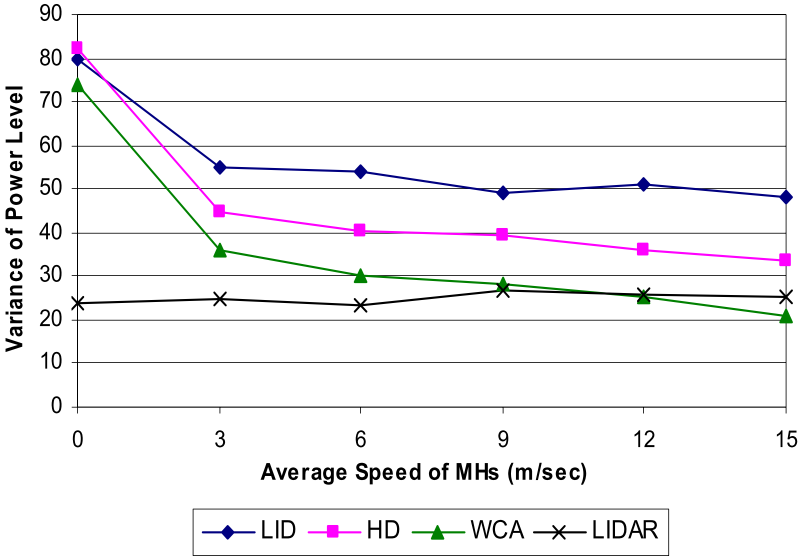
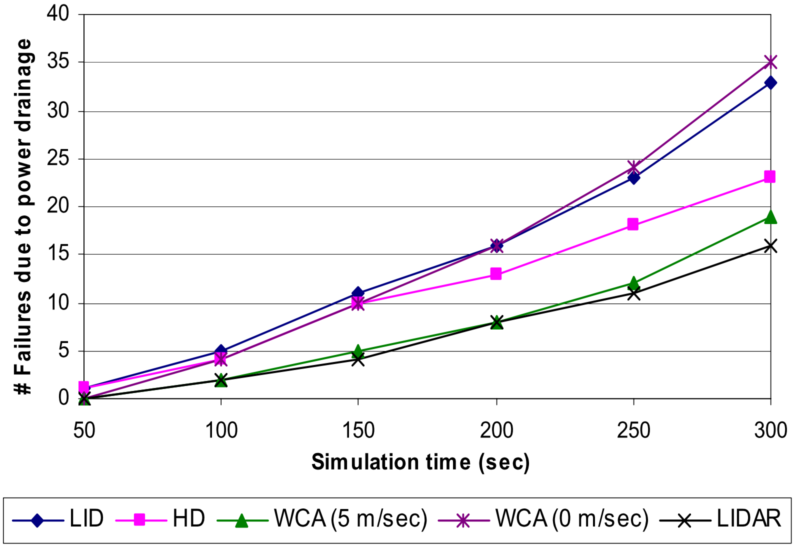


Fig. 9 Number of node failures due to energy depletion (for 200 MHs) during the simulation time



Finally, an interesting aspect of our proposed algorithm's performance is revealed in Figs. 10 and 11. In particular, those figures indicate a trade-off among cluster stability (CH changes) and node failures (due to energy depletion) related to the values of weight coefficients of (1); hence, a compromise among these two performance parameters is required.

5 Conclusions—future work

In this article, we have introduced a novel protocol that explicitly separates clustering process in cluster formation and cluster maintenance phases. The former extends the ideas of LID algorithm increasing the likelihood for electing CHs with low mobility and sufficient energy capacity. The latter aims at minimizing control overhead and enhancing cluster stability, without sacrificing the balanced consumption of energy supplies among MANET nodes.

The main contributions of our protocol, compared to existing solutions, are summarized in the following: (a) clustering formation is completed in a fast and cost-inexpensive manner; (b) both mobility and battery power metrics are taken into account in clustering formation, so that suitable nodes are elected as CHs and energy consumption is uniformly distributed among network nodes; (c) we have introduced a simple, novel method for measuring the mobility rate, which poses no requirement for GPS card mounted on MHs; (d) control traffic volume is greatly reduced during clustering maintenance phase, while preventing CHs energy depletion occurrences; (e) fast packet forwarding and delivery is enabled, as clusters are pro-actively formed and topol-

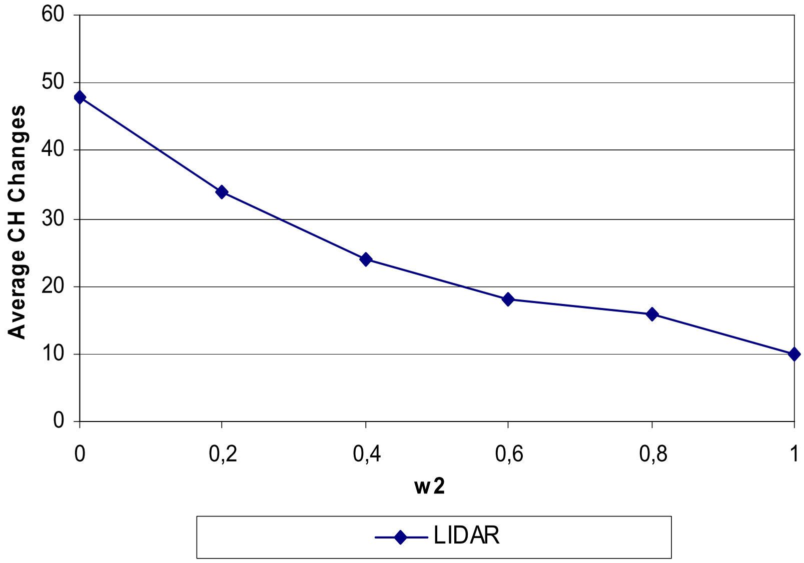


Fig. 10 Average number of CH changes in LIDAR algorithm as a function of mobility metric weight coefficient (for 200 MHs)

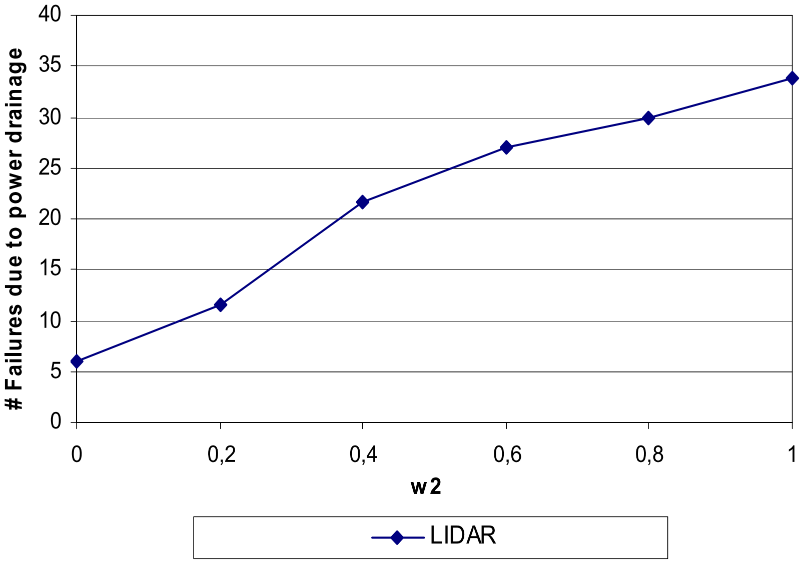


Fig. 11 Average number of node failures occurrences in LIDAR algorithm as a function of mobility metric weight coefficient (for 200 MHs)

ogy information is available when actual user data exchange is required.

Simulation results demonstrated that LIDAR protocol outperforms traditional LID and HD algorithms, as well as a more recent approach (WCA) in terms of control traffic overhead, cluster stability, variance of energy level among MHs and number of node failures due to battery drainage.

As a future extension, we will investigate the incorporation of a mobility prediction method, e.g. similar to [22], to identify group mobility patterns and provide steadier cluster formations. The effect of MHs transmission range in the operation of LIDAR will be evaluated for all typical ranges of the standard 802.11a equipment [8]. We also intend to extend our cluster maintenance algorithm so as to restrict the number of nodes dominated by a single CH between a lower and an upper bound; that way, clusters will be small enough to impede drainage of CHs resources and large enough to prevent long routing paths and message delivery delays. A variation of the LIDAR algorithm where node IDs are received, sorted and re-assigned by a single (centralized) node will also be implemented and compared in terms of cost against the distributed ID re-assignment method described in this article.

Acknowledgements The research work presented herein has been co-funded by 75% from EU and 25% from the Greek government under the framework of the Education and Initial Vocational Training II, Programme Archimedes.

References

1. An, B., & Papavassiliou, S. (2001). A mobility-based clustering approach to support mobility management and multicast routing in mobile ad-hoc wireless networks. *International Journal of Network Management*, *11*(6), 387–395.
2. Baker, D. J., Ephremides, A., & Flyn, J. A. (1984). The design and simulation of a mobile radio network with distributed control. *IEEE Journal on Selected Areas in Communications*, *2*(1), 226–237.
3. Basagni, S. (1999). Distributed and mobility-adaptive clustering for multimedia support in multi-hop wireless networks. In *Proceedings of the 50th IEEE vehicular technology conference* (pp. 889–893).
4. Belding-Royer, E. M. (2002). Hierarchical routing in ad hoc mobile networks. *Wireless Communications and Mobile Computing*, *2*(5), 515–532.
5. Chatterjee, M., Das, S. K., & Turgut, T. (2002). WCA: a weighted clustering algorithm for mobile ad hoc networks. *Cluster Computing*, *5*, 193–204.
6. Chen, Y. P., Liestman, A. L., & Liu, J. (2004). Clustering algorithms for ad hoc wireless networks. In Y. Pan & Y. Xiao (Eds.), *Ad hoc and sensor networks*. Nova Science Publishers.
7. Chiang, C. C., Wu, H. K., Liu, W., & Gerla, M. (1997). Routing in clustered multihop, mobile wireless networks with fading channel. In *Proceedings of the IEEE Singapore international conference on networks (SICON'97)* (pp. 197–211).
8. Cisco (2004). *Cisco aironet 1230AG series 802.11a/b/g access point data sheet*. Cisco Systems.
9. Ephremides, A., Wieselthier, J. E., & Baker, D. J. (1987). A design concept for reliable mobile radio networks with frequency hopping signalling. In *Proceedings of the IEEE, 75* (pp. 56–73).
10. Gavalas, D., Pantziou, G., Konstantopoulos, C., & Mamalis, B. (2006). Lowest-ID with adaptive ID reassignment: A novel mobile ad-hoc network clustering algorithm. In *Proceedings of the 1st IEEE international symposium on wireless pervasive computing*.
11. Gavalas, D., Pantziou, G., Konstantopoulos, C., & Mamalis, B. (2006). Clustering of mobile ad hoc networks: An adaptive broadcast period approach. In *Proceedings of the IEEE international conference on communications* (pp. 4034–4039).
12. Gavalas, D., Pantziou, G., Konstantopoulos, C., & Mamalis, B. (2006). Stable and energy efficient clustering of wireless ad-hoc networks with LIDAR algorithm. In *Lecture notes in computer science: Vol. 4217. Proceedings of the 11th IFIP International Conference on Personal Wireless Communications (PWC'2006)* (pp. 100–110).
13. Gavalas, D., Pantziou, G., Konstantopoulos, C., & Mamalis, B. (in press). ABP: A low-cost, energy-efficient clustering algorithm for relatively static and quasi-static MANETs. *International Journal of Sensor Networks*.
14. Guptar, P., & Kumar, P. R. (2000). The capacity of wireless networks. *IEEE Transactions on Information Theory*, *IT-46*(2), 388–404.
15. Gerla, M., & Tsai, J. T. C. (1995). Multicluster, mobile, multimedia radio network. *Wireless Networks*, *1*(3), 255–265.
16. Hong, X., Xu, K., & Gerla, M. (2002). Scalable routing protocols for mobile ad hoc networks. *IEEE Network*, *16*(4), 11–21.
17. Li, C. R., & Gerla, M. (1997). Adaptive clustering for mobile wireless networks. *IEEE Journal of Selected Areas in Communications*, *15*(7), 1265–1275.
18. Li, F., Zhang, S., Wang, X., Xue, X., & Shen, H. (2004). Vote-based clustering algorithm in mobile ad hoc networks. In *Lecture notes in computer science: Vol. 3090. Proceedings of International Conference on Networking Technologies for Broadband and Mobile Networks* (pp. 13–23).
19. McDonald, B., & Znati, F. (1999). A mobility-based framework for adaptive clustering in wireless ad hoc networks. *IEEE Journal on Selected Areas in Communications*, *17*, 1466–1487.
20. McDonald, B., & Znati, F. (2001). Design and performance of a distributed dynamic clustering algorithm for ad-hoc networks. In *Proceedings of the 34th annual simulation symposium* (pp. 27–35).
21. Perkins, C. (2001). *Ad hoc networking*. Addison-Wesley.
22. Sivavakeesar, S., Pavlou, G., & Liotta, A. (2004). Stable clustering through mobility prediction for large-scale multihop ad hoc networks. In *Proceedings of the IEEE wireless communications and networking conference*.
23. Yu, J., & Chong, P. (2005). A survey of clustering schemes for mobile ad hoc networks. *IEEE Communications Surveys*, *7*(1), 32–48.
24. Xu, K. X., Hong, X. Y., & Gerla, M. (2002). An ad hoc network with mobile backbones. In *Proceedings of the IEEE international conference on communications* (pp. 3138–3143).
25. Zheng, R., & Kravets, R. (2003). On-demand power management for ad hoc networks. In *Proceedings of the IEEE Infocom'2003* (pp. 481–491).

Damianos Gavalas received his BSc degree in Informatics (Computer Science) from the University of Athens, Greece, in 1995 and his MSc and Ph.D. degree in electronic engineering from University of Essex, U.K., in 1997 and 2001, respectively. Currently, he is an Assistant Professor in the Department of Cultural Technology and Communication, University of the Aegean, Greece. His research interests include distributed computing, mobile code, network and systems management, e-commerce, mobile ad-hoc and sensor networks.

Grammati Pantziou received the Diploma in Mathematics and the Ph.D. Degree in Computer Science from the University of Patras, Greece, in 1980 and 1991, respectively. She was a Post-Doctoral Research and Teaching Fellow at the University of Patras (1991–1992), a Research Assistant Professor at Dartmouth College, Hanover, NH, USA (1992–1994), an Assistant Professor at the University of Central Florida, Orlando, FL, USA (1994–1995) and a Senior Researcher at the Computer Technology Institute, Patras (1995–1998). Since September 1998, she is a Professor at the Department of Informatics of the Technological Educational Institution of Athens, Greece. Her current research interests are in the areas of parallel computing, design and analysis of algorithms, distributed and mobile computing and multimedia systems.

Charalampos Konstantopoulos received the Diploma (five year first degree) in computer engineering from the Department of Computer Engineering and Informatics at University of Patras, Greece (1993). He also received his Ph.D. degree in Computer Science from the same department in 2000. Currently, he is a Ph.D. researcher at the Research Academic Computer Technology Institute, Patras, Greece. His research interests include parallel and distributed algorithms/architectures and mobile computing.

Basilis Mamalis obtained his diploma (in Computer Engineering and Informatics) as well as his Ph.D. from the University of Patras (Greece), in 1993 and 1997 respectively. He has worked for almost ten years (in several projects and research programs) in Computer Technology Institute (C.T.I.) of Patras as well as an academic teacher in the Hellenic Open University. During the last two years he teaches as an Assistant Professor at the Technological Educational Institute (T.E.I.) of Athens, Dept. of Informatics. His research interests include information retrieval, document and knowledge management, parallel algorithms and architectures, distributed systems and databases, operating systems and mobile computing.