

# Stable and Energy Efficient Clustering of Wireless Ad-Hoc Networks with LIDAR Algorithm

Damianos Gavalas<sup>1</sup>, Grammati Pantziou<sup>2</sup>, Charalampos Konstantopoulos<sup>3</sup>, Basilis Mamalis<sup>2</sup>

<sup>1</sup> Department of Cultural Technology and Communication, University of the Aegean, Greece, dgavalas@aegean.gr

<sup>2</sup> Department of Informatics, Technological Education Institute of Athens, Greece, {pantziou, vmamalis}@teiath.gr

<sup>3</sup> Computer Technology Institute, Patras, Greece, konstant@cti.gr

**Abstract.** The main objective of clustering in mobile ad-hoc network environments 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. A popular clustering algorithm (LID) suggests CH election based on node IDs (nodes with locally lowest ID value become CHs). Although fast and simple, this method is biased against nodes with low IDs, which are likely to serve as CHs for long periods and are therefore prone to rapid battery exhaustion. Herein, we propose LIDAR, a novel clustering method which represents a major improvement over traditional LID algorithm: 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. 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.

## 1. Introduction

Current wireless cellular network infrastructures rely on a wired backbone connecting base stations, implying that networks are fixed and constrained to a geographical area with a pre-defined boundary. Deployment of such networks takes time and cannot be set up in times of utmost emergency. Therefore, mobile multi-hop radio networks, known as mobile ad hoc networks (MANETs), play a critical role in places where a wired (central) backbone is neither available nor economical to build, such as law enforcement operations, battle field communications, disaster recovery situations, and so on [15]. Such situations require a dynamic network topology where all nodes, including routers, are mobile and communication between two end nodes can be supported by intermediate nodes.

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 encounter scalability problems with increased network size, especially in the face of node mobility [10]. 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* [3]. Clustering not only makes a large MANET appear smaller, but more importantly, it makes a highly dynamic topology to appear less dynamic [12].

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*. 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 [1][2][6][8][9][11]. 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 [10]. However, some clustering schemes employ explicit message exchange among MHs in *periodic* basis for maintaining the cluster structure [8][9][11]; 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) [1][2]. 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 [4]. 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.

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 [11], 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.

The remainder of the paper is organized as follows: Section 2 overviews related work and explains the motivation for our research. Section 3 describes the details of our proposed protocol, while Section 4 discusses simulation results. Finally, Section 5 concludes the paper and draws directions for future work.

## 2. Related Work & Motivation

Several heuristics have been proposed to address ad-hoc networks clustering problem. One of the most popular ones is the Lowest-ID (LID) [11], 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 [9], 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 [2].

Vote-based clustering (VC) [8] 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 [8] revealed that the inclusion of the degree metric certainly affects clusters stability, similarly to HD 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 [8]. 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 [17]. 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) [2] 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, article [2] 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 [17].

### 3. Description of our Proposed Protocol

In this article, we propose a novel clustering protocol, Lowest-ID with Adaptive ID Reassignment (LIDAR). LIDAR 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 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 [18]. 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 [3]. 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 [12][16]. 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 [6][8][17]. 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:

$$W_v = w_1 B_v - w_2 M_{v,t}, \quad w_1 + w_2 = 1 \quad (1)$$

where  $B_v$  denotes the remaining battery life of node  $v$  and  $M_{v,t}$  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).

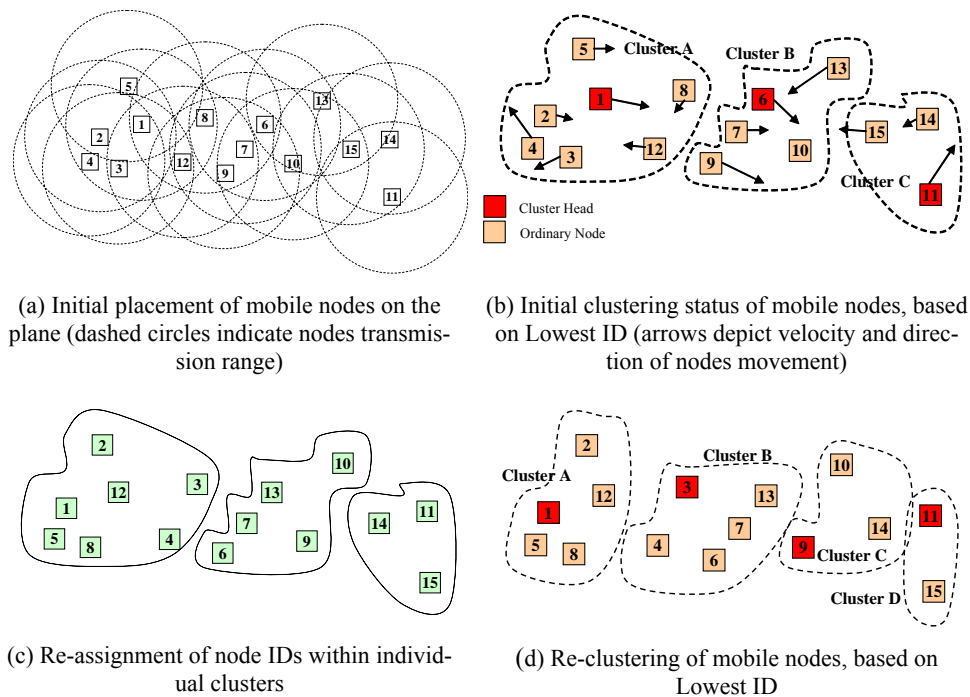
**Step 3:** Whenever re-clustering is needed (in the following section we discuss the circumstances under which re-clustering process is triggered), CHs request their attached MHs to send their  $W_v$  values through a special broadcast message (WEIGHT\_REQUEST).

**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.

**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).



**Fig. 1.** Illustration of LIDAR execution steps

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. 1. Table 1. presents how  $W_v$  values are calculated, where the coefficients of equation (1) are set to  $w_1 = 0.7$  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

**Table 1.** Calculation of  $W_v$  values and node IDS re-assignment in LIDAR (where  $w_1 = 0.7$  and  $w_2 = 0.3$ )

Most existing methods for estimating nodes mobility rate pose the requirement for GPS card with sufficient accuracy mounted on every mobile node. We propose an alternative method for measuring mobility rate which relaxes mobile nodes from such requirement. In particular, each CH measures its neighborhood mobility rate through contrasting the topology information it obtains during successive HPs.

A main objective of LIDAR algorithm is to minimize control traffic overhead during clustering formation phase, which highly depends on HP duration (i.e. frequency of broadcasting ‘Hello’ control packets). To achieve that, CHs measure the mean mobility rate of their attached cluster members MRC and accordingly adapt the ‘Hello’ broadcast period BP within their cluster. It is also guaranteed that HP duration always lies between two boundaries:  $HP_{\min} \leq HP \leq HP_{\max}$ ; at startup, HP is globally set to  $HP_{\min}$ . The details of our mobility rate measurement method may be found in [7].

### 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 [10]. 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 [13]. 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 [17].

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  $B_{CH} < T * \frac{\sum_{v=1}^d B_v}{d}$  (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 [2], 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 should also be 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.

**(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 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. 1.d, 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 topol-

ogy 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 then 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. Simulation Results

LIDAR protocol has been simulated using NS-2 simulator [14] 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.

A square terrain of  $600\text{m} \times 600\text{m}$  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 - 15m/s, on random direction. At the event of reaching the terrain boundary, MHs are bounced back. The ‘hello period’ duration is set to 1 sec 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’.

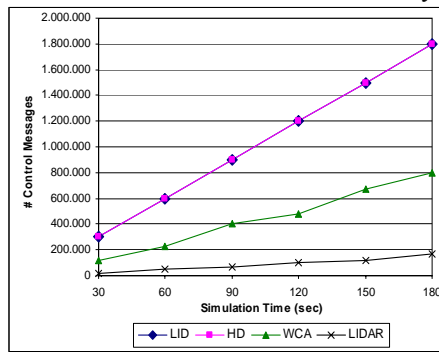
Fig. 3a illustrates the average number of control messages exchanged as simulation time advances. In LID and HD algorithms, ‘Hello’ messages are periodically broadcasted during cluster maintenance phase; hence, their performance results coincide. WCA executes re-clustering whenever a MH moves to a region not covered by any CH [2]. 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.

Fig. 3b 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. In contrast, LIDAR’s per-

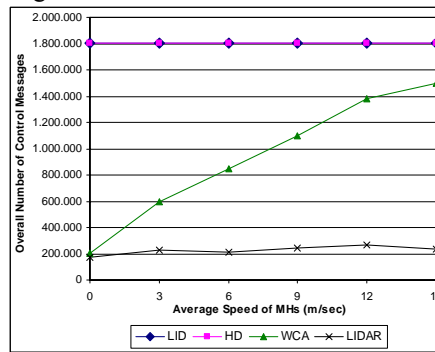


formance remains unaffected; yet, it depends on the frequency of transmission requests.

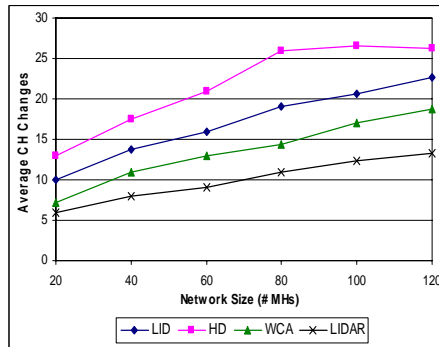
Fig. 3c compares the average number of CH changes, which is an indicator of the overall cluster structure stability (the more frequent the CH changes, the less stable clusters are). As expected, LID performs better than HD as the former exclusively uses ID and the latter node degree information to decide 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.



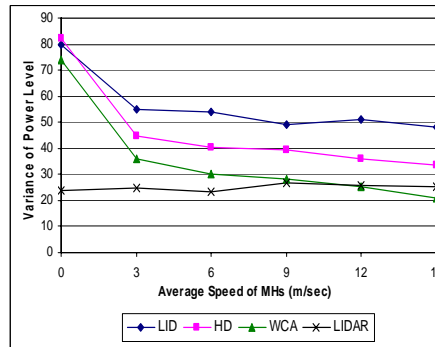
(a) Average number of control messages during simulation runs (50 MHs, with average speed of MHs 5 m/sec and average rate of transmission requests for MHs 1 request/min).



(b) Overall number of control messages (50 MHs, with average rate of transmission requests for MHs 1 request/min).



(c) Average number of CH changes (for average speed of 5 m/sec and average rate of transmission requests for MHs 1 request/min).



(d) Variance of energy level among MHs (50 MHs, with average rate of transmission requests for MHs 1 request/min).

**Fig. 2.** Simulation results

Finally, Fig. 3d 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 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.

## **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.

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 and variance of energy level among MHs.

As a future extension, we intend to incorporate mobility metric in the calculation of weight function values, and also introduce a mobility prediction method (e.g. similar to [16]) 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 [5]. 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.

## **Acknowledgments**

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] S. Basagni, "Distributed and Mobility-Adaptive Clustering for Multimedia Support in Multi-Hop Wireless Networks," Proceedings of the 50th IEEE Vehicular Technology Conference (VTS'99), pp. 889–93, September 1999.
- [2] M. Chatterjee, S. K. Das, D. Turgut, "WCA: A Weighted Clustering Algorithm for Mobile Ad Hoc Networks", Cluster Computing, 5, pp. 193–204, 2002.
- [3] Y. P. Chen, A. L. Liestman, J. Liu, "Clustering Algorithms for Ad Hoc Wireless Networks", in "Ad Hoc and Sensor Networks" (ed. Y. Pan and Y. Xiao), Nova Science Publishers, 2004.
- [4] C.-C. Chiang et al., "Routing in Clustered Multihop, Mobile Wireless Networks with Fading Channel" Proceedings of IEEE SICON'97, pp. 197-211, 1997.
- [5] "Cisco Aironet 1230AG Series 802.11a/b/g Access Point Data Sheet", Cisco Systems 2004.
- [6] D. Gavalas, G. Pantziou, C. Konstantopoulos, B. Mamalis, "An Efficient and Scalable Clustering Algorithm of Wireless Ad Hoc Networks", Proceedings of the 1<sup>st</sup> International Workshop on Distributed Algorithms and Applications for Wireless and Mobile Systems (DAWMS'2005), pp. 761-766, November 2005.
- [7] D. Gavalas, G. Pantziou, C. Konstantopoulos, B. Mamalis, "Lowest-ID with Adaptive ID Reassignment: A Novel Mobile Ad-Hoc Network Clustering Algorithm", Proceedings of the 1st IEEE International Symposium on Wireless Pervasive Computing (ISWPC'2006), January 2006.
- [8] F. Li, S. Zhang, X. Wang, X. Xue, H. Shen, "Vote-Based Clustering Algorithm in Mobile Ad Hoc Networks", Proceedings of International Conference on Networking Technologies for Broadband and Mobile Networks (ICOIN'2004), LNCS vol. 3090, pp. 13 – 23, February 2004.
- [9] M. Gerla, J.T.C. Tsai, "Multicluster, mobile, multimedia radio network", Wireless Networks 1(3), pp. 255–265, 1995.
- [10] X. Hong, K. Xu, M. Gerla, "Scalable Routing Protocols for Mobile Ad Hoc Networks", IEEE Network, 16(4), pp. 11-21, July-Aug, 2002.
- [11] C. R. Li, M. Gerla, "Adaptive Clustering for Mobile Wireless Networks", IEEE Journal of Selected Areas in Communications, 15(7), pp. 1265-1275, September 1997.
- [12] B. McDonald, F. Znati, "A Mobility-Based Framework for Adaptive Clustering in Wireless Ad Hoc Networks", IEEE Journal on Selected Areas in Communications, Vol. 17, pp. 1466 –1487, August 1999.
- [13] B. McDonald, F. Znati, "Design and Performance of a Distributed Dynamic Clustering Algorithm for Ad-Hoc Networks," Proceedings of the 34th Annual Simulation Symposium, pp. 27–35, April 2001.
- [14] Network Simulator - NS-2, <http://www.isi.edu/nsnam/ns/>.
- [15] C. Perkins, "Ad Hoc Networking", Addison-Wesley, January 2001.
- [16] S. Sivavakeesar, G. Pavlou, A. Liotta, "Stable Clustering Through Mobility Prediction for Large-Scale Multihop Ad Hoc Networks", Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC'2004), IEEE, March 2004.
- [17] J. Yu, P. Chong, "A Survey of Clustering Schemes for Mobile Ad Hoc Networks", IEEE Communications Surveys, 7(1), pp. 32-48, March 2005.
- [18] R. Zheng, R. Kravets, "On-demand Power Management for Ad Hoc Networks," Proceedings of the IEEE Infocom'2003, pp. 481–91, Mar.–Apr. 2003.