Ad Hoc Networks 81 (2018) 17–44

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

Energy-aware backbone formation in military multilayer ad hoc networks

D. Papakostas[a], S. Eshghi[b], D. Katsaros[b,*], L. Tassiulas[b]

[a] Department of Electrical & Computer Engineering, University of Thessaly, Volos, Greece
[b] Department of Electrical Engineering & Yale Institute for Network Science, Yale University, USA

ARTICLE INFO

Article history:
Received 14 August 2017
Revised 17 June 2018
Accepted 24 June 2018
Available online 9 July 2018

Keywords:
Energy awareness
Minimum connected dominating set
Backbone formation
Multilayer networks
Wireless ad hoc networks
Military networks

ABSTRACT

Ad hoc networks designed for deployment in modern battlefields need to take care of traditional requirements related to their backbone's size, their energy efficiency, their scalability in terms of network size, but also of their nature which allows for combining networks of different units that act altogether towards a common operational goal. This article develops a distributed algorithm for developing an energy-aware backbone for military ad hoc network composed of multiple layers, namely *E2CLB*. The algorithm is based on the concepts of connected dominating sets and also on node centrality concepts, and results as a heuristic solution to the problem of calculating a maximum energy, minimum connected dominating set for a multilayer network by including into the dominating set those nodes which are highly connected to their and other layers (i.e., they have large centrality value) and moreover they are energy-rich. The computation and communication complexities of the algorithm are analyzed, and a thorough simulation-based evaluation of it against six competitors is presented. The results show that E2CLB is either the best performing algorithm across the examined performance measures or it is able to trade a very small increase in the size of the backbone's network in order to reap improved performance in the energy realm.

1. Introduction

A military ad hoc network is a type of ad hoc network which encompass some unique characteristics compared to a traditional wireless ad hoc network [1,2]. Apart from the broadcast-nature of the wireless communication medium and mobility which are very common features, these ad hoc networks are usually very large in terms of the number of participating nodes. Therefore, more critically than for 'plain' ad hoc networks, we need to ensure protocol scalability in the number of nodes. Moreover, we must carefully consider for reduced delays and for the scarce energy resources. Additionally, due to the dynamic topology, protocols must be based on primitives that are feasible and efficient to compute in a distributed manner, and also to engage only computations based on localized information. Most important though is to consider the nature of the network itself which usually consists of "subnetworks". For instance, Tactical wireless networks built with the Joint Tactical Radio System (JTRS) have layers of subnets; these subnets are built up with waveforms. There is the soldier radio waveform (SRW) tier. It can have two subtiers, one for soldier-to-soldier communications and one for networking sensors. Above that, there is the wideband networking waveform (WNW) tier, which has two subtiers; one forms local subnets for vehicle-to vehicle communications, and the other is for global connectivity, to generate a single subnet over the entire theater. There is also the Joint Airborne Network-Tactical Edge (JAN-TE) stub network that supports the tactical airborne domain of weapons platforms.

We consider 'island' subnetworks as being the layers of a single, large network, which we call a multilayer communication network. To make clear this nature, we show in Fig. 1 a mixed military unit consisting of a tank platoon belonging to some tank company, and an infantry squad belonging to some infantry platoon. These two units communicate wirelessly via an ad hoc network and advance in the battlefield pursuing some common operational goal.

In this wireless network we would recognize two "subnetworks", namely the tank layer and the soldiers' layer; for various reasons related to military strategy and hierarchy and terrain topology, the only links are those shown in the figure. Earlier methods did not allow a node to participate into two "networks" at the same time, but recent progress in networking could sustain such situations. In [3] we used terminology from complex networks literature to describe the topology of such ad hoc networks, and we will use here the same terminology. Thus, we rec-

* Corresponding author.
E-mail addresses: papdimit@inf.uth.gr (D. Papakostas), soheil.eshghi@yale.edu (S. Eshghi), d.katsaros@yale.edu, dkatsar@inf.uth.gr (D. Katsaros), leandros.tassiulas@yale.edu (L. Tassiulas).

https://doi.org/10.1016/j.adhoc.2018.06.017
1570-8705/

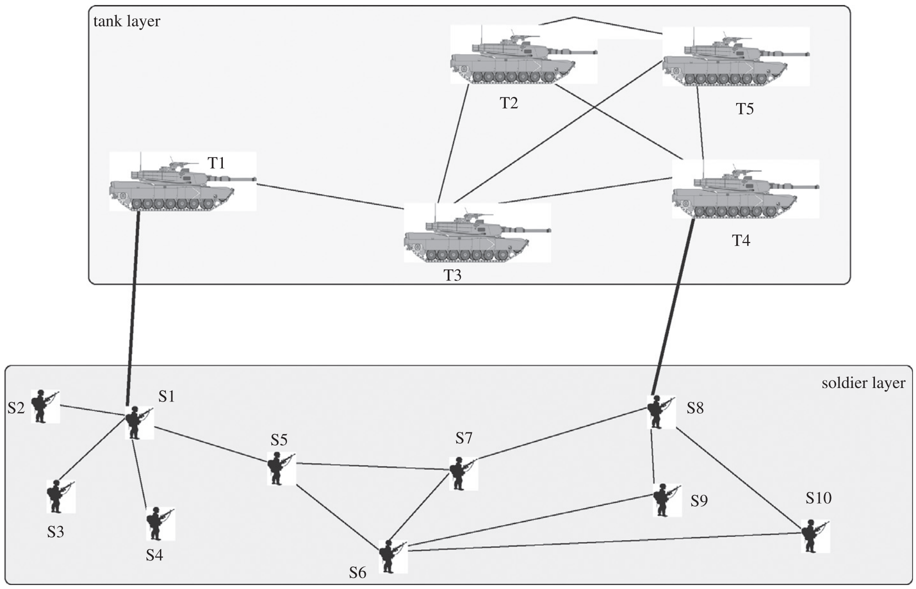

Fig. 1. Abstraction of a military multilayer ad hoc network comprised by 2 layers. Physical obstacles have been removed, and the entities have been projected onto the 2D space.

ognize two network *layers, intralayer* connections (the thin ones) connecting entities of the same type, and *interlayer* links (the tick ones) providing connections among entities belonging to different layers.

As in any wireless ad hoc network, the construction of a backbone network is a fundamental goal in order to provide other, higher level operations. The two most prevalent, scalable solutions for that goal are backbone construction protocols based on *node clustering* and protocols based on *dominating sets*. The former approach [4,5] will not work efficiently because of node mobility, and even clustering protocols for vehicular ad hoc networks [6] are not appropriate because they assume that mobility of nodes takes place strictly along roads. On the other hand, connected dominating set-based backbone construction protocols [7,8] are robust, flexible solutions, and thus we consider them as the preferred method to work with in the present work. Diverse types of military units participate in modern battlefields that are energy-constrained, such as soldier, drones, still sensors, and so on. Therefore, backbone construction protocols must be energy-aware. In this paper, we investigate the topic of energy-aware backbone construction for military multilayer networks using connected dominating sets constructed in a distributed fashion. In principle, any efficient algorithm for calculating a minimum connected dominating set seeks to detect nodes strategically positioned in the topology in order to include them into the dominating set and thus decrease the size of the obtained dominating set, because they 'dominate' over a large number of other nodes. For instance, some algorithms for single layer networks achieve this by looking at the degree of each node [9]. Moreover, if while searching for nodes to include in the dominating set we include as criterion apart from their strategic position, their residual energy, then we can develop energy-aware algorithms for dominating set construction.

1.1. Motivation and contributions

The literature on dominating set-based backbone construction is very rich and spans more than two decades; however the architecture of multilayer networks poses some new challenges. Firstly, the existence of layers demands a different treatment than considering each layer in isolation or damping layer information and applying existing algorithms. It was proved in [3] that solutions based on decomposition and/or aggregation of the entire multilayer network are not efficient; there is significant room for improvement if we take into account the multilayer structure. Secondly, assigning different weights on intralayer versus interlayer links can not help transform our problem at hand into that of dealing with the calculation of a weighted dominating set of the multilayer network, because there is no algorithmic method yet for the determination of the relative weights so as to produce an efficient backbone for multilayer networks. Finally, energy-related issues have not been investigated for dominating set-based backbone construction algorithms for multilayer networks, even though there is work on energy-agnostic protocols [3].

In this context, the present article makes the following contributions:

- it investigates the issue of energy-aware connected dominating set-based backbones for multilayer networks, and it generalizes an earlier proposed centrality measure for identifying nodes with high residual energy and central position within the multilayer network;
- it develops a distributed algorithm, namely *E2CLB* which is based on the aforementioned centrality measure for identifying dominating nodes;
- it analyzes the algorithm's performance both from a computational/communication complexity perspective and an experimentation-based perspective, and it it compares exhaustively the proposed algorithm against relevant and baseline competitors, because there is no prior work on the article's subject.

The rest of this article is organized as follows: Section 2 introduces in formal terms the problem of constructing energy-aware dominating sets for multilayer networks; Section 3 proposes a locally computable measure to assess the significance of a node in participating in an energy-aware connected dominating set; Section 4 develops a distributed algorithm for calculating the energy-aware connected dominating set; Section 5 provides performance evaluation results comparing the proposed algorithm against competitors; Section 6, briefly presents related works, and finally Section 7 concludes the article.

2. Problem formulation

The main elements of the architecture of a multilayer ad hoc network are the following: the network consists of a set of nodes, each one belonging to some layer, and having an amount of energy associated with it which is described by a scalar quantity. Each node has a non-empty set of connections towards (some) nodes belonging to the same layer (intralayer links), and it has a (possibly empty) set of connections towards nodes belonging to other layers (interlayer links). All links are assumed to be bidirectional.

We will now describe our setting using graph-theoretic terms. A multilayer network which consists of n layers is a pair (G^{ML}, E^{ML}), where $G^{ML} = \{G^i, i = 1, \ldots, n\}$ is a set of 'networks' (G_i, E_i) ($|G_i|$ nodes belonging to layer i, and $|E_i|$ edges connecting nodes belonging to layer G_i), and a set of interlayer links $E^{ML} = \{E_{i,j} \subseteq G_i \times G_j; i, j \in \{1, \ldots, n\}, i \neq j\}$. Moreover, each node is annotated with a scalar quantity which represents its residual energy. Then, the problem of finding an energy-aware backbone network based on dominating sets for multilayer networks in a distributed fashion can be described as follows:

Definition 1 (The ML-MEMCDS problem)**.** The problem of calculating a *Maximum Energy Minimum Connected Dominating Set* for a multilayer network (G^{ML}, E^{ML}) consists of finding a subset *MEMCDS* of its nodes such that the following conditions hold:

1. Each node of G^{ML} either belongs to *MEMCDS* or is adjacent to (in one hop distance from) a node belonging to *MEMCDS*.
2. The cardinality of set *MEMCDS* is the minimum possible.
3. The nodes comprising *MEMCDS* are connected to each other, i.e., there is path from any node $i \in MEMCDS$ to any node $j \in MEMCDS, \forall i, j$. [Intralayer or interlayer links may comprise that path.]
4. The sum of energies of nodes belonging to *MEMCDS* is the maximum possible.
5. Knowledge of only the closest k-hop neighborhood of a node is permitted.

Corollary 1. *The problem ML-MEMCDS is NP-complete.*

The proof is trivial [10] and thus we omit it. Versions of the problem with directed links, with incremental maintenance of its solutions in cases of nodes/links additions/removals will be examined in subsequent articles.

We proved in [3] (Theorem 1) that finding a minimum connected dominating sets for every layer and then trying to connect them does not provide efficient solutions in terms of minimizing the cardinality of the dominating set. Similar observations were made in [3] for methods based on ignoring the layer information and calculating connected dominating sets in the 'aggregated' network. It is easy to extend those results for our case where energy issues are present. Thus, in the next two sections, we will present an efficient heuristic solution to this problem that considers the layering information.

3. Identifying energy-rich cross-layer relay nodes

In this section we will introduce a locally computable measure to identify prominent nodes to be included in the *MEMCDS*. For the sake of article's completeness we will first give some useful definitions from previous works.

Definition 2 (Power Community Index (PCI) [11])**.** The *PCI* index of a node v is the maximum number k, such that there are k 1-hop neighbors of this node with degree larger than or equal to k.

PCI coincides with the well-known h-index [12]. We have extended this for the case of multilayer networks:

Definition 3 (Minimal-layers PCI (mlPCI_n) [3,13])**.** The $mlPCI_n(v)$ index of a node v is the maximum number k, such that there are at k direct (1-hop) neighbors of v with the number of links *towards* n different layers greater than or equal to k.

mlPCI_n characterizes a node for its connectivity in a predefined number of layers. We further combine $mlPCI_n$ values for all n, thus defining *mlPCI* as follows:

$$mlPCI(v) = \sum_{i=1}^{\#layers} mlPCI_i(v). \quad (1)$$

mlPCI categorizes as 'good' nodes those who are well connected in many layers compared to those who are well connected in a few layers.

A disadvantage of the original PCI (and thus of *mlPCI*) is that it is mainly based on the connectivity of the nodes that participate in the definition of *PCI*; the connectivity of the rest of the nodes is ignored. We should somehow incorporate this missed topological information into our definitions. We do this for a single layer as follows: we calculate the *PCI* index of a node as usual (using Definition 2) and then – after excluding the nodes that contributed to this *PCI* value – we compute a new *PCI* value with the remaining nodes, and add the two *PCI* values. We perform this computation for every layer, and add the resulting indices; we call the obtained number *Exhaustive PCI* (*xPCI*). *xPCI* is not satisfactory as a ranking mechanism because it creates a lot of ties. To this end, for those k nodes that participate in each *PCI* index, we calculate the number of unique links between them in order to form the final index. Actually, we multiply each *PCI* value by log_2 of the number of links in order to obtain reasonable values for our measure even for large networks, and also to let nodes with quite similar connectivity to get very similar values. We call this new measure *Cross-layer PCI* (*clPCI*).

In [3] we developed a backbone construction algorithm based on clPCI which was energy-agnostic, i.e., all nodes were assumed to somehow replace the energy they deplete, e.g., by fuel, solar panels, etc. However, in the generic case, energy issues do need to be considered for ad hoc connectivity, especially in battlefields [14]. Thus, we provide here a simple generalization of *clPCI*, namely *EclPCI*which, for a node u with energy equal to $E(u)$ is defined as follows:

$$EclPCI(u) = E(u) \times clPCI(u). \quad (2)$$

When energy is not an issue, then clearly $EclPCI \equiv clPCI$. Algorithm 1 presents a distributed algorithm for the calculation of *EclPCI*of node u.

Note that because *EclPCI* is calculated on a *per layer* basis it is possible to present some sort of preference to one or more layers. For example, if the multilayer network incorporates a layer with nodes with no energy issues, then it might be a wise decision to have many relay nodes in that layer. This capability however is out of the scope of the present work and it will not be examined further. In case of ties due to Eq. (2), the selection of relay nodes may be random, or in an application-dependent way, e.g., preferring energy-rich nodes over well connected for resource-scarce environments.

Proposition 1. *The computation complexity of EclPCIindex calculation is* $O(\Delta^2)$ *in the worst case, where* Δ *is the maximum node degree in the network.*

Proof. The worst case regarding the computation complexity of the *EclPCI* index calculation is when a host u has Δ neighbors and each one of them has Δ neighbors too; i.e. $PCI(u) = \Delta$. In such case and during the calculation of the unique links among neighbors, a host u needs to compare its neighbor set with Δ neighbors and the neighbor set comparison has a complexity of $O(\Delta)$. □

Algorithm 1: *EclPCI* index value calculation.

precondition : Known 1-hop ($N(u)$) and 2-hop ($N^2(u)$) neighbor connectivity info (ID) of node u
postcondition: Calculation of the *EclPCI* index value of node u
remarks : m = number of layers in the multilayer network, $layer(u)$ = network layer that node u is situated, $E(u)$: residual energy of node u, S = node set, $PCI(u)$, $xPCI(u)$, $clPCI(u)$, $EclPCI(u)$: indexes related to node u

for *layer* $i \leftarrow 1$ **to** m **do**
  $PCI(u) = xPCI(u) = 0$;
  Build $S = u_1, u_2, \ldots, u_n \mid u_{k\,(1\leq k\leq n)} \in N(u), layer(u_{k\,(1\leq k\leq n)}) = i$;
  while $S \neq empty$ **do**
    Calculate $PCI(u)$ *for* S;
    Calculate unique links ($Links_{unique}$) of nodes participating in $PCI(u)$;
    $xPCI(u)$ += $PCI(u)$ * $\log_2(Links_{unique})$;
    Remove nodes that participated in $PCI(u)$ from S;
    $PCI(u) = Links_{unique} = 0$;
  end
  $clPCI(u)$ += $xPCI(u)$;
end
$EclPCI(u) = E(u) * clPCI(u)$;

Energy-related augmentation can be applied to *mlPCI*as well, and in that case we get the *EmlPCI*measure. Now, armed with a method to identify energy-rich nodes whose connections span many nodes in many layers, we are ready to describe a distributed algorithm for calculating an energy-aware connected dominating set.

4. Distributed energy-aware formation algorithm

*EclPCI*which was described in previous section is actually a *centrality* measure that identifies those nodes of the network which have high energy levels and at the same time maintain a strategic/central position among the network layers. In principle, any efficient heuristic algorithm for calculating a minimum connected dominating set seeks to detect such strategically positioned nodes in order to decrease the size of the obtained dominating set. Some algorithms for single layer networks achieve this by looking at the degree of each node [9].

Thus, we exploit the *EclPCI* measure and incorporate it into a distributed algorithm for computing an energy-aware connected dominating set. The algorithm will be called *Energy-awarE Cross-Layer Backbone* formation algorithm, and in the sequel we will use the key *E2CLB*for it. In principle, a backbone based on the formation of a connected dominating set whose elements are such well-connected nodes can turn them into hotspots. There are several solutions proposed in the literature [15] that can alleviate these kinds of problems, e.g., role rotation, movement control and so on; in general it is a well addressed problem and therefore we will refrain from replicating the details of such mechanisms here. Additionally, we need to mention that making *E2CLB*able to work for unidirectional links, or weighted links (e.g., using the weight to depict variation in energy consumption during communication) is straightforward by simply incorporating direction/weight in the calculation of *EclPCI*. Such adaptations are abundant in the literature for centrality measures [16], and thus we skip the relative discussion here. Finally, node location is implicitly but firmly taken into account by *EclPCI*via the selection of links; it is an abstract and thus generic mechanism avoiding the use of raw geographical coordinates that do not facilitate change of systems of reference.

Before delving into technical details of the proposed algorithm, we will first provide an brief overview of the algorithm, and then we will describe its constituent parts. In *E2CLB*, there are mainly three phases, which are the following: (1) CDS construction, (2) redundant relay node pruning, and (3) mediator phase. Before these three steps take place, one more procedure evolves that is typical and common in (almost) all distributed algorithms for ad hoc networks with non GPS-enabled nodes. During this process, each node learns the topology of its neighborhood, and also other interesting features (e.g. residual energy) of its neighbors. For *E2CLB*, each node learns the connectivity and residual energy of all its neighbors up to its 2-hop neighborhood $N^2(u)$; this preparatory phase will not be described in details since it is very common.

The CDS construction phase is based on a *source-initiated* relay node selection process that is executed by every node u. Because this selection process produces many redundant CDS nodes, a pruning phase follows. Finally, in order to exploit the connectivity among nodes that belong to the same relay node set and improve the minimum residual energy level of a node in the set, one more phase called the mediator phase is employed which is based on some heuristic rules.

4.1. CDS construction phase

CDS construction (Algorithm 2) is divided into two tasks, namely *neighbor prioritization* and *construction* task. During neighbor prioritization task, every node u calculates its own *EclPCI* index and it broadcasts its value in a single message to its neighbors. By mutuality of the distributed protocol, it receives its neighbors' *EclPCI* values. Then, it sorts the nodes in $N(u)$ in non-increasing order of their *EclPCI* value. In the construction stage, each node u selects from $N(u)$ and includes in its relay node set $R(u)$ the nodes with the largest *EclPCI* index value that cover at least one new node in the $N^2(u)$ neighborhood. Using the proof methodology

Algorithm 2: Relay node set election.

precondition : Known 1-hop ($N(u)$) and 2-hop ($N^2(u)$) neighbor connectivity info (ID) of node u
postcondition: Elected relay node set ($R(u)$) of node u
remarks : $EclPCI(u)$: index related of node u, $M(u)$: status of node u with regards to being [True (T)]or not [False (F)] a relay node

Calculate and broadcast own *EclPCI* index value;
Gather the *EclPCI* index values of the nodes in $N(u)$;
Sort nodes in $N(u)$ in decreasing order of their *EclPCI* index values;
repeat
  Select the node from $N(u)$ with the largest *EclPCI* index value that covers at least one new node in $N^2(u)$;
  Include the selected node in $R(u)$;
until *each node in* $N^2(u)$ *has at least one neighbor in* $N(u)$;
Broadcast $R(u)$;
if *selected as a relay node and* $M(u) = F$ **then**
  $M(u) = T$; `/* node becomes a relay node */`
  Broadcast status change;
end
Update 1-hop neighborhood node status (if req);

of [17, Theorem 4.2], we can easily prove that the resulting relay node sets of all the network nodes form a CDS.

Proposition 2. *The computation complexity of the relay node set election process is $O(\Delta^3)$, where Δ is the maximum vertex degree in the network.*

Proof. The prioritization phase involves neighbor sorting based on *EclPCI* value, which is a $O(\Delta * log\Delta)$ operation. The worst case regarding the construction phase results when a host u has Δ neighbors and each one of them contributes Δ nodes to the coverage of the 2-hop neighborhood of u. In this case, host u needs to run once over its neighbor set of size $O(\Delta)$ and 'erase' those nodes of the 2-hop neighborhood of u (which has maximum size $O(\Delta^2)$) covered by the specific neighbor; therefore, this operation costs $O(\Delta^3)$, i.e., the total cost progresses as follows: $\Delta^2 + (\Delta^2 - \Delta) + (\Delta^2 - 2\Delta) + \dots + (\Delta^2 - (\Delta - 1)\Delta)$. □

4.2. Pruning phase

It is known that distributed, source-initiated dominating set construction algorithms produce dominating sets with many redundant nodes [18,19]. For our needs, we design a distributed pruning phase, which is executed by relay nodes only. Each candidate relay node is aware of its status, i.e., being a relay or not due to step 8 of Algorithm 2. Each relay node waits until all its one-hop neighbors decide their 'relay status' before it enters the pruning phase (Algorithm 3).

Moreover, in order to confront the case where more than one relay nodes enter the pruning phase simultaneously, we "prioritize" the execution of the pruning rules in such a way that relay nodes with smaller residual energy level execute it earlier than relay nodes with larger residual energy level. In order to do that each node i whenever is selected as a relay node it calculates a backoff time according to Eq. (3):

$$T_{pruning} = \frac{E(i)}{|R(u)|} + \ell, \tag{3}$$

Algorithm 3: The pruning phase.

precondition : Completed relay node set election process from 1-hop neighbors
postcondition: Node updated status
remarks : $T_{pruning}$: a timer, $S_{constrained}$: a node set, $M(u)$: status of node u with regards tobeing [True (T)] or not [False (F)] a relay node

Start $T_{pruning}$;
Build $S_{constrained} = u_1, u_2, \dots, u_n \,|\, u_{k\,(1 \le k \le n)} \in N(u) \wedge N^2(u)$, $M(u_{k\,(1 \le k \le n)}) = T$, $EclPCI(u) < EclPCI(w_{k\,(1 \le k \le n)})$;
if $S_{constrained}$ *is subject to* $N(u) \subset N(u_1) \cup N(u_2) \dots \cup N(u_n)$ *and* $u_1, u_2, \dots, u_n$ *form a connected graph* **then**
  Wait for expiration of $T_{pruning}$;
  if $M(u_{k\,(1 \le k \le n)}) = T$ **then**
    $M(u) = F$; `/* node becomes a plain node */`
    Broadcast status change;
    Exit pruning stage;
  else
    Restart pruning stage;
  end
else
  $M(u) = T$; `/* node remains a relay node */`
  Broadcast status;
  exit pruning stage;
end

$E(i)$ is the residual energy of node i, and $|R(u)|$ is the cardinality of the relay node set of node u that node i participates in; ℓ is a unique pseudo-random number calculated by node i in the range [0, 0.1] that is used in order to solve any ties between relay nodes in $|R(u)|$ with the same residual energy level.

So, each relay node before starting to execute the pruning rules waits first for the backoff time to expire. In the case where more than one nodes have selected the same node i as a relay node, that relay node will calculate more than one backoff times, i.e., a backoff time of each different relay node set that i participates in, but use during the pruning stage only the smaller backoff time between them. It is interesting to notice that the backoff time formula favors the elimination of relay nodes that have either small residual energy and/or belong to a relay node set with many participants. To achieve a good balance between efficiency and overhead in our work we make use of the *restricted pruning Rule k* as this self-pruning scheme, in general, is more efficient in reducing the relay node set than several existing schemes that ensure the broadcast coverage [20]. This rule can be implemented with knowledge of either 2-hop or 3-hop neighborhood [21]. (In the appendix, we provide results concerning the performance of both alternatives.) In the pruning rule we make use of connectivity as quantified by *EclPCI* as priority value in order to establish a total order among nodes that participate in the CDS. Connectivity has been proved to be the most efficient priority under all circumstances [20]. The complete pruning phase is depicted in Algorithm 3.

Proposition 3. *The computation complexity of the pruning phase is $O(\Delta^3)$, where Δ is the maximum vertex degree in the network.*

Proof. A relay node u in order to decide if it will act as a relay node or not it needs to calculate the coverage capability of a connected graph composed of both 1-hop and 2-hop neighbors. Thus, each relay node u compares its neighbor set with Δ^2 neighbors in the worst case, and the neighbor set comparison has a $O(\Delta)$ complexity. □

4.3. The mediator phase

The central idea of this phase is to further reduce the relay node set by examining if a particular relay node can be accessed through another relay node; we call this relay node a *mediator* (Algorithm 4). The mediator heuristic is employed sequentially to relay nodes of the same set, in increasing order of their *EclPCI* value. Thus, a relay node i with smaller *EclPCI* index value than other nodes from the same relay node set will be examined first if it can be reached through another relay node, and if so, it will be removed from the respective relay nodeset *iff* it has less residual energy from the relay node that will act as a mediator.

Moreover, with the intention of avoiding race conditions regarding a relay node that is included in more than one relay node sets we resorted to prioritizing the removal of the relay nodes in such a way that nodes who have smaller *EclPCI* index value take higher priority to decide about their relay node sets than other nodes that have larger *EclPCI* index value. In order to do that, each node u calculates a backoff time and executes the mediator heuristic right after the expiration of the respective $T_{mediator}$ timer. The mediator backoff time is calculated with Eq. (4).

$$T_{mediator} = \frac{EclPCI(u)}{|R(i)|}. \tag{4}$$

$E(u)$ is the residual energy of node u and $|R(i)|$ is the cardinality of the relay node i that is under consideration to be removed (it is used for normalization purposes). All in all, the mediator heuristic is an indirect approach to sustain as long as possible the number of alive nodes in the network [22,23] or equally the fraction of

Algorithm 4: The mediator phase.

precondition : Completed pruning process from 1-hop relay nodes
postcondition: Updated relay node set
remarks : $T_{mediator}$: backoff timer, S_{relays}: node set, $R(u)$: relay node set of node u, $M(u)$: status of node u with regards tobeing [True (T)] or not [False (F)] a relay node

Start $T_{mediator}$;
Update $R(u)$ = $u_1, u_2, \ldots, u_n \mid u_{k\,(1 \leq k \leq n)} \in N(u), M(u_{k\,(1 \leq k \leq n)}) = T$;
Sort nodes in $R(u)$ in increasing order of their *EclPCI* index value;
Broadcast $R(u)$;
Set S_{relays} = $R(u)$;
Sort nodes in S_{relays} in increasing order of their residual energy;
Wait for expiration of $T_{mediator}$;
repeat ∀ node $v_{k\,(1 \leq k \leq n)}$ in $R(u)$, in increasing order of their *EclPCI* index value
    repeat ∀ node $w_{k\,(1 \leq k \leq n)}$ in S_{relays}, in increasing order of their residual energy level
        if $E(w_k) > E(v_k)$ *and* $v_k \in R(w_k)$ **then**
            remove v_k from $R(u)$;
            Broadcast $R(u)$;
            Set w_k as a mediator to get to v_k;
            break;
        else
            select the next node from S_{relays};
        end
    until *Until all nodes in* S_{relays} *are checked*;
    select the next node from $R(u)$;
until *Until all nodes in* $R(u)$ *are checked*;

alive nodes [24]. Next, we present the pseudocode of the mediator heuristic.

Proposition 4. *The computational complexity of the mediator phase is* $O(\Delta^2 \times \log\Delta)$ *in the worst case, where* Δ *is the maximum degree in the network.*

Proof. In the worst case, a node with degree equal to Δ will have Δ relays. Thus, after sorting them (with cost $\Delta \times \log\Delta$) a serial scan over them takes place with cost $O(\Delta)$ and while scanning each, a new sorting over the rest relays is performed with cost $O(\Delta \times \log\Delta)$. □

4.4. Communication overhead of E2CLB

The following theorem presents the communication overhead and latency (in terms of information exchange) of the proposed algorithm.

Proposition 5. *In bidirectional networks, the execution of E2CLB algorithm requires* 7 *rounds to complete.*

Proof. The 2-hop information used by the relay node set election process can be collected via two rounds of information exchanges. In round 1, each node advertises its *ID* and residual energy level and builds its 1-hop neighbor set based on the advertisement of its neighbors. In round 2, each node advertises its 1-hop neighbor set and identifies links among 1-hop neighbors. These two rounds are present in any distributed protocol where the nodes need to become aware of their neighborhood. In round 3, each node calculates its *EclPCI* index value and advertises it together with its 2-hop neighbor set. Then it identifies links among 2-hop neighbors. In round 4, each node calculates and advertises its own relay node set and updates 1-hop neighbor status. In round 5, the restricted Rule k is applied to each relay node and each one of them advertises its status. In round 6 each node advertises its updated relay node set and applies the mediator heuristic to each one of the participating relay nodes. Finally, in round 7 the composition of the updated relay node set is advertised (if needed). □

5. Performance evaluation

In this section we will present the details of the evaluation setting and illustrate the results. In particular, in Section 5.1 we present the competing algorithms, and in Section 5.2 we give the performance measures of the comparison. In Section 5.2.1 we describe the network topologies used in our simulation, and in Section 5.3, we present and comment on the obtained results.

5.1. Competing algorithms

The first thing to note is that, instead of *EclPCI*, we can use in its position the *EmlPCI*measure and thus get the *Energy-awarE MultiLayer Backbone* formation algorithm (*E2MLB*); or we can use the *clPCI*measure – which does not take the residual energy of a node into account – and get the *Energy Unaware Cross Layer Backbone* formation algorithm (*EUCLB*) which is actually the algorithm proposed in [3]. These two algorithms along with some baseline ones that will be described in the next paragraph will be used as competitors to *E2CLB*.

Degree-based CDS construction is a very popular technique, and thus we looked for generalizations of degree centrality in multilayer networks. We call the respective competitor as *E2WDB* which uses a generalized notion of degree found in [25]. This algorithm uses the same mechanics as *E2CLB* to create the CDS; in particular it uses 2-hop connectivity information and it incorporates the pruning phase. However, in its plain version it does not include the mediator heuristic, but its enhanced version called *E2WDB*⋆ does include this heuristic.

The next competing algorithm is based on Tang et al. algorithm to form a MCDS [26], namely *EMCDS*. This algorithm is not localized, as it requires global information to compute the relay node set. However, it can produce a near-optimal forward node set. Here we use it as a substitution of a “perfect” algorithm that produces the optimal result both in terms of the size of the CDS constructed and the energy efficiency. It is emphasized that *EMCDS*is based on an 1-hop connectivity info in order to build the CDS. The impact of the mediator heuristic on the performance of *EMCDS*is presented separately under the algorithm *EMCDS*⋆.

5.1.1. Analytic computation and communication complexity of the competitors

Apparently, *E2MLB* and *EUCLB* present the same communication overhead with *E2CLB*, because they use the same heuristics during the CDS construction. On the other hand, *EMCDS*communication overhead varies according to the position of the most energy efficient node in the network [26]. To detect this in a distributed fashion, we need $O(n^* logn)$ [27] messages by constructing some spanning tree, and then we need $O(diameter)$ rounds for termination where each node sends $O(1)$ messages. Therefore, *EMCDS* has $O(n^* logn)$ message complexity, and $O(diameter)$ delay. *E2MLB*and *EUCLB*present the same computation complexity with *E2CLB; E2WDB*variations and *EMCDS*variations have $O(\Delta)$ cost.

Table 1
Experimentation parameters values.

Parameter	Range	Default
Avg. node degree (D)	4, 6, 10, 12, 16	6
Network diameter (H)	5, 10, 20, 40, 70	10
#network layers (L)	2, 3, 4, 5, 7	4
Size of a layer relative to its adjacent layers	10%, 20%, 30%, 50%, 70%	-

5.2. Performance measures

So far the evaluation of an energy-aware construction algorithm in a routing protocol-independent way is done according to one of the following ways: (i) the first node to die, (ii) the number (or fraction) of alive nodes, (iii) the time until the network fails to construct a backbone, (iv) the fraction of connected dominating set nodes that remain alive, (v) the time until the packet delivery ratio drops "drastically". In this work we employ several detailed – and not simply gross – generalized performance measures which are described in the sequel. Competing algorithms are compared in terms of the size of the CDS, the mean *per node* minimum node energy in the relay set, the mean cardinality of each relay node set, and the message complexity to build each relay node set. We say an algorithm is more efficient than another algorithm if it generates a smaller CDS [8,26]. Additionally, an algorithm that manages to establish *per node* a relay set with larger minimum residual energy level is considered to be more energy efficient than another algorithm whose *per node* relay set includes relay nodes with less residual energy; this measure is a direct approach to define the network lifetime. Moreover, we use the size of the relay set as another performance measure, as the smaller the relay set *per node*, the smaller the volume of broadcast message transmissions in the network is, which subsequently translates into a reduction in node interference, bandwidth usage, and energy savings for the non-relay nodes.

5.2.1. Datasets

Due to the lack of publicly available, real world military multilayer networks, we developed a generator for multilayer weighted networks in MATLAB. Our aim was to build a generator that could create in an algorithmic way a variety of multilayer weighted network topologies. The generator should be able to generate topologies where the degree of a node, the diameter of each layer, the size of each layer, and the number of layers could vary after defining some parameters. The generator was developed and described in detail in [13], but here we will present its basic features.

There are several wireless testbeds, e.g., NITOS[1] and several emulation environments for ad hoc networking research [28]. However, the disadvantage of all of them is that they only allow for experimentation with networks consisting of a few dozens of nodes. On the contrary, the requirement of modern battlefields is to able to operate ad hoc networks consisting of twenty-fold more nodes; for instance a battalion would need a thousand nodes.[2] Thus, we opted out of performing small scale experiments, and worked with a default setting that allowed for four layers consisting of 500 nodes each (see Table 1), and experimented with even larger network sizes, e.g., with seven layers.

So in our topologies each network layer consists of a set of wireless nodes distributed in a two-dimensional plane. Each node has the same *maximum* transmission range *R*. By proper scaling, we set that all nodes have the same maximum transmission range equal to one. Every pair of nodes whose Euclidean distance is equal to or less than this maximum transmission range are assumed to be connected, i.e., they form a *Unit Disc Graph* (UDG). So in this way the actual location of nodes is taken into account when computing the connectivity. Moreover, in order to better approach reality where obstacles prohibit the direct communication between adjacent nodes, we used non-uniform intra-layer models to distribute the nodes on the two-dimensional plane, the same way it was done in [29]. The construction of a multilayer network is controlled by the link density in each layer which is expressed by the average degree of each node, by the number of nodes per layer (i.e., size of the layer), and the number of layers.

The task of interconnecting the different layers was done with the aid of two parameters: the number of links a node has towards nodes in different layers, while the second parameter involves the distribution of interconnections towards the nodes within a certain layer. Finally, we want to have control on the way energy (i.e., weights) are distributed among nodes. Given the above considerations, we apply the *Zipfian* distribution for our interconnectivity generator which can produce from uniform to highly skewed distributions for every parameter of interest. The desired skewness is managed by parameter $s \in (0, 1)$. We apply four distinct *Zipfian* distributions, one per parameter of interest:

- $s_{degree} \in (0, 1)$ in order to generate the frequency of appearance of highly interconnected nodes,
- $s_{layer} \in (0, 1)$ in order to choose how frequently a specific layer is selected,
- $s_{node} \in (0, 1)$ in order to choose how frequently a specific node is selected in a specific layer.
- $s_{weight} \in (0, 1)$ in order to choose how much uniformly weights are distributed in the multilayer network.

We use two different approaches to apply the *Zipfian* laws; i.e., by selecting nodes either in increasing or decreasing order of their degree. We selected a default setting for each of the parameters of interest and created various datasets that we used to evaluate the efficiency of each competing algorithm. Collectively, we call these parameters as the *topology skewness*, and represent it as a sequence of four floats, e.g., $0.5 - 0.5 - 0.5 - 0.5$, meaning that $s_{degree} = 0.5$, $s_{layer} = 0.5$, $s_{node} = 0.5$ and $s_{weight} = 0.5$ (which are the default settings we used to create the datasets). We perform experiments and present the performance of the competing algorithm when using datasets which differ in the topology skewness settings. In a multilayer network the relative size of the layers clearly has an impact on the performance of the algorithms. Thus, we equipped our topology generator with the ability to create multilayer topologies where each layer can be a percentage (10%, 20%, 30%, 50%, 70%) larger than the previous one. So we have topologies with relatively equi-sized layers (10%), or topologies with huge layer inequalities (70%). Table 1 records all the independent parameters of our topology generator, their range of values, and their default values.

5.3. Simulation results

We performed a simulation-based performance evaluation of the competing algorithms in MATLAB. We repeated each experiment 5 times, and recorded the variation in the performance, but each result was so tightly concentrated around the mean that the error bars are hardly recognizable in the plots.

5.3.1. Impact of topology density

Throughout this section, we consider the impact of topology density on the performance of each competitor. Firstly, in Fig. 2 we evaluate the *per layer* size of the CDS that each competitor creates. The first observation is that the size of the CDS is almost a

[1] https://nitlab.inf.uth.gr/NITlab/nitos.
[2] https://www.darpa.mil/news-events/2013-04-30.

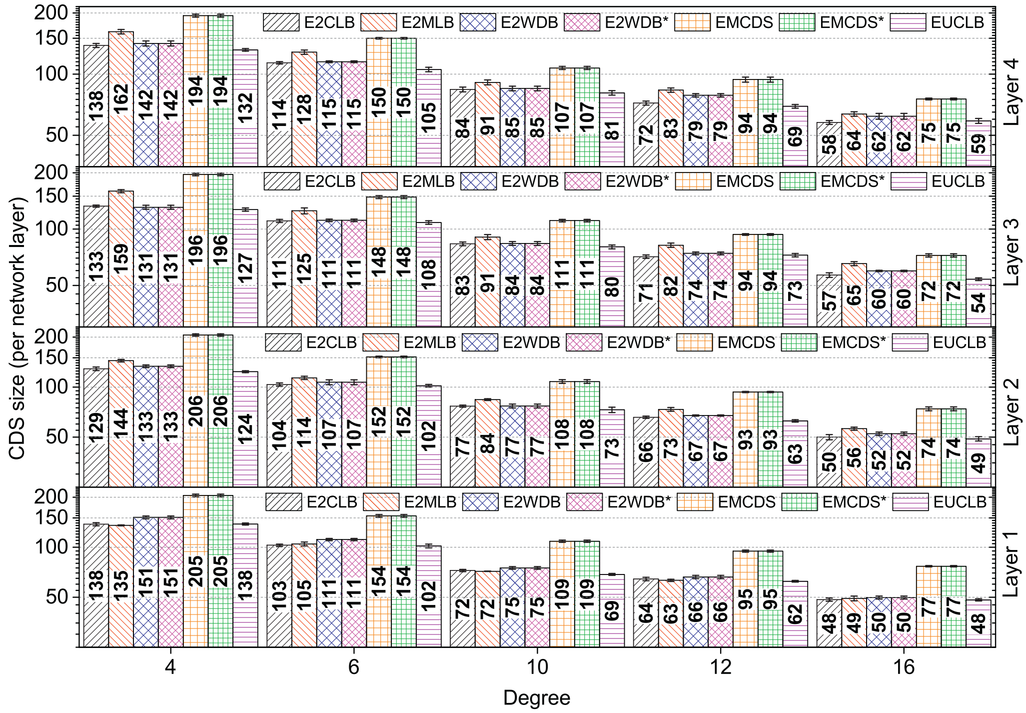

Fig. 2. Impact of network density on the size of CDS.

decreasing function with respect to the node density, which is consistent with the existing results previously obtained in [30]. That is due to the fact that the higher the network density the greater the coverage capability of the multilayer network nodes, and thus the smaller the size of the CDS. It is interesting that the distribution of the CDS nodes among the layers is almost uniform for *EMCDS* (up to 5% variance) and for both *E2CLB, EUCLB* (up to 10% variance), while it increases in each layer for *E2MLB* (approximately from 5% up to 10%). The aforementioned behavior has to do with the different way that each competitor creates the CDS. In *EMCDS*, each node that is selected to participate in the CDS, selects recursively its own nodes for the CDS which result to the uniform distribution of the CDS nodes on the multilayer network nodes.

On the other hand, both *EUCLB* and *E2CLB*calculate per layer the CDS. The unique behavior of *E2MLB* is justified by the fact that it multiplexes different layers in order to calculate the *EmlPCI* value. In EMCDS the size of the CDS is from 27% (best case) up to 60% (worst case) larger than the best performing algorithm, which is *EUCLB*. The high performance of *EUCLB* regarding the size of the CDS has been shown in [3]. The second best performing algorithm is *E2CLB*(generally, both algorithms present almost the same performance, but in some cases *E2CLB*presents up to 9% worse performance e.g., at layer 4 when degree $= 6$), third is *E2WDB* (up to 10% worse performance) and then follows *E2MLB* (up to 20% worse performance). Focusing on all competitors, we observe that the difference in their performance is minimum when degree $= 16$. This is due to fact that when nodes are relatively “close” to each other, there is significant overlapping in the selected CDSs. *E2WDB*⋆ and *EMCDS*⋆ performance is not considered here as the mediator heuristic does not affect the total number of CDS nodes in the network. Therefore these two improved algorithms present the same efficiency regarding the size of the CDS with their “clean” versions.

Next, in Fig. 3 we evaluate the *mean per layer node* minimum relay node energy. The first observation is that compared to the previous experiment, now *EUCLB* presents the worst performance (in most cases). That is expected as *EUCLB* is unaware of the residual energy of each of the selected nodes for the CDS. However, we see that in some cases *EUCLB* presents even better performance than *EMCDS* does (e.g., when degree $= 4$), but this is due to the smaller CDS it creates.

The second observation is that generally the competitors create *per layer node* more efficient CDS as the network density increases. This is due to the fact that the larger network density presents more opportunities for nodes with smaller residual energy level to be substituted by more energy efficient nodes. The best performing algorithms are *E2CLB* and *E2MLB* (we examine the performance of *E2WDB*⋆ and *EMCDS*⋆ right afterwards), with the first being from 4% (when considering relatively sparse networks) up to 20% (when considering relatively dense networks) more efficient than the second one. The performance gap when considering networks with different density is due to the fact that in dense networks both the pruning process and the mediator stage work more efficiently; i.e., in dense networks it is more likely to find nodes with less energy and exclude them from the CDS or reach them through other nodes which have a larger residual energy level. The third best performing algorithm is *EMCDS* and last comes *E2WDB*. However, *E2WDB* performs better than *EMCDS* when degree $= 4$, which is justified by the fact that *EMCDS* creates a large CDS (more than 35% larger than the CDS of *E2WDB*) and consequently many nodes with less energy are likely to participate in the CDS. This however does not exist in denser network topologies (except for the Layer 1 when degree $= 6, 10, 16$). The next observation has to do with the fact that the mediator heuristic is very efficient. Both *E2WDB*⋆ and *EMCDS*⋆ present better performance compared to their version that lacks the heuristic. More specifically, *E2WDB*⋆ is from 4% (when considering relatively sparse networks) up to 21% (when considering relatively dense networks) more efficient than *E2WDB*. For *EMCDS*⋆, the corresponding figures are better compared to *EMCDS* from 9% up to 24% (in most cases is even better than *E2MLB* when degree > 4). The mediator heuristic is more effective in *EMCDS*⋆ because it creates a relay node set with larger cardinality, thus the likelihood to be removed those nodes that participate in the CDS and have less residual energy increases.

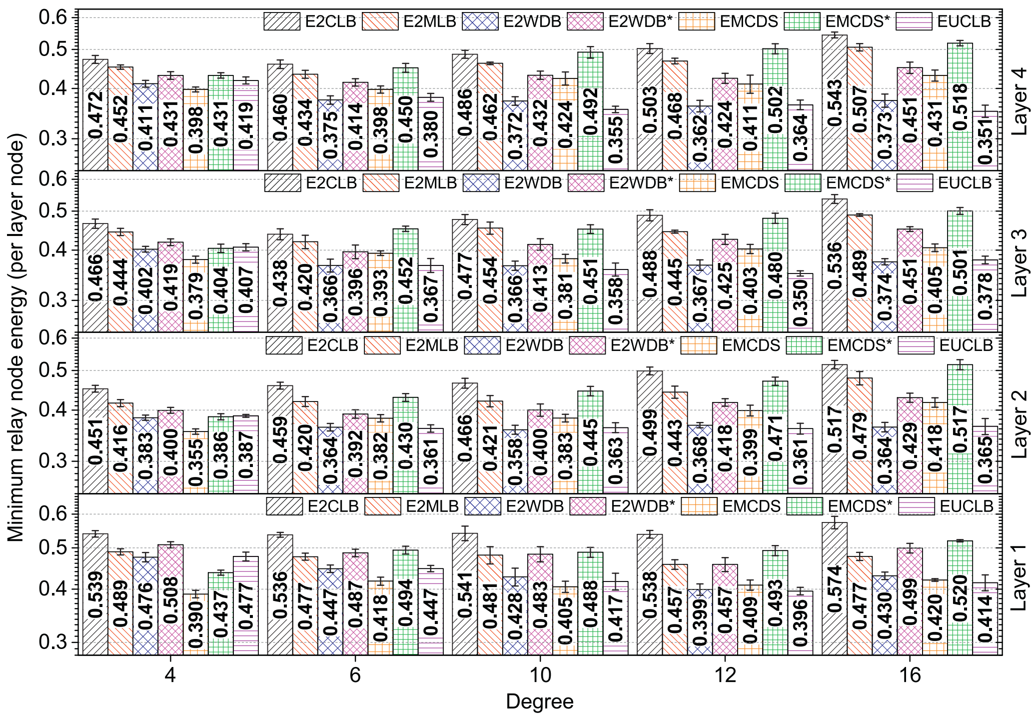

Fig. 3. Impact of network density on the energy level of each relay node (on the average the worst case scenario).

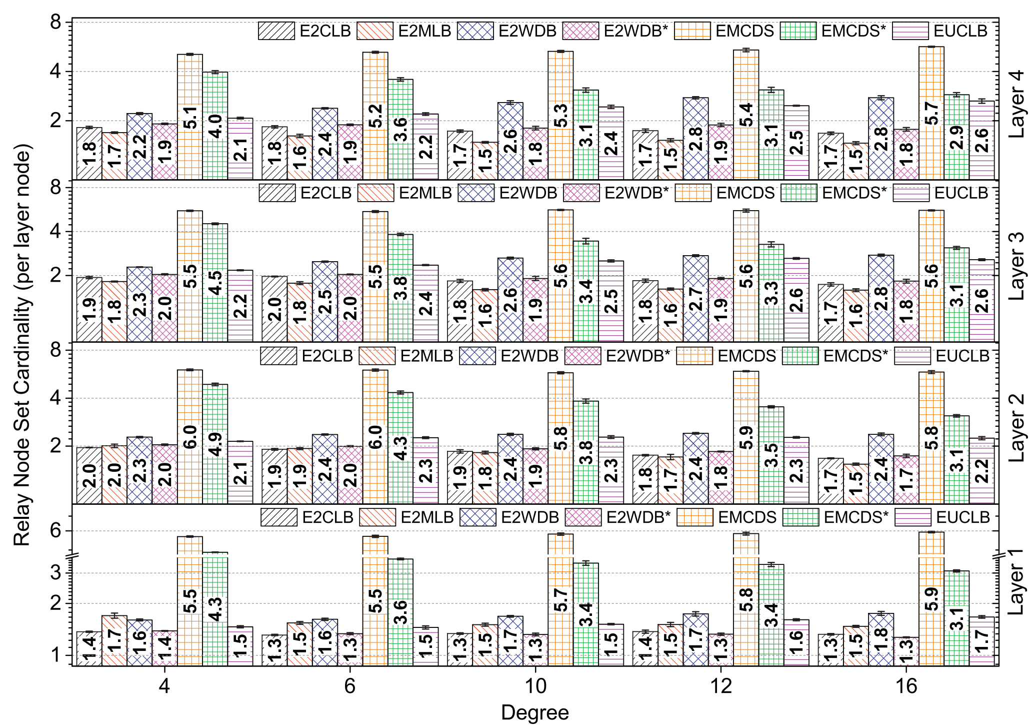

Fig. 4. Impact of network density on the size of the relay node set of each network node (on the average).

Next, in Fig. 4 we evaluate the *mean per layer node* relay node set cardinality. The first observation is that all competitors (except from *EMCDS* and *EMCDS*⋆ which use a different approach in order to calculate the CDS) create small *per layer node* relay sets. This is something desirable in order to reduce the number of redundant messages in broadcasting situations [19]. The best performing algorithm is *E2MLB* (which interestingly is presenting the larger CDS) and then follows *E2CLB*, *E2WDB*⋆, *E2WDB* and finally *EUCLB* (which presents the smaller CDS). The second observation is that the mediator heuristic for one more time improves the efficiency of *E2WDB* and *EMCDS* regarding the *per layer node* relay node set cardinality (on the average) by 14% up to 55% for the *E2WD** and by 28% up to 90% for the *EMCDS*⋆. The higher efficiency of the mediator heuristic in *EMCDS*⋆ is justified by the larger CDS that *EMCDS* produces compared to *E2WDB*.

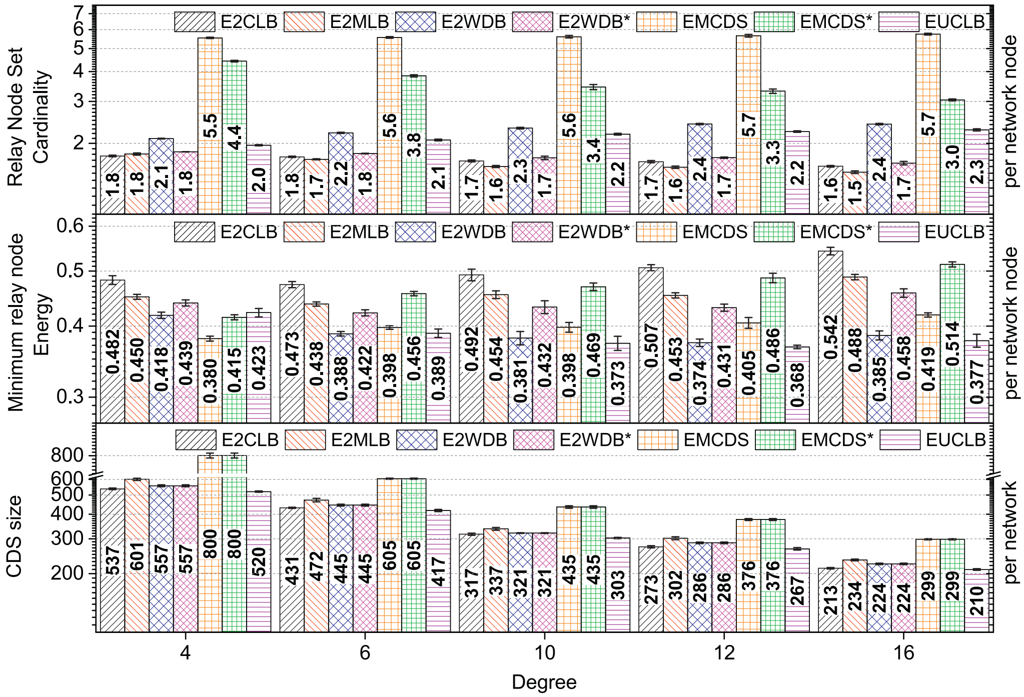

Fig. 5. Impact of network density on the performance of each algorithm.

Finally, in Fig. 5 we summarize the aforementioned *per layer* results and present them in one diagram in order to have a better overview of the impact of the topology density on the performance of each competitor. From the bottom plot we conclude that the size of the CDS decreases as the network density increases, for every algorithm. From the middle plot we conclude that generally the CDS efficiency (in terms of the minimum energy that each CDS node has) is proportional to the network density. Finally, from the upper plot we conclude that the *per network node* size of the relay node set is irrespective to the network density.

5.3.2. Impact of network diameter

In this section, we consider the impact of network diameter on the performance of each competitor. Firstly, in Fig. 6 we evaluate the *per layer* size of the CDS that each competitor creates. The first observation is that as the network diameter increases the size of the constructed CDS for all algorithms increases. The increment of the CDS is the result of sparser vicinities, i.e., fewer links between the network nodes. In other words, fewer, longer (in hops), and less distinct paths exist towards the nodes of the multilayer network, which renders the election of those nodes that compose the backbone and ensure the overall network connectivity less discrete, and hence more nodes are recruited.

It is interesting that while all the competitors manage to keep the *per layer* size of the CDS under control for the various diameter parameter settings (from 5% up to 20% CDS increase per diameter setting until when diameter = 40), they fail to do that when diameter = 70 and the *per layer* size of the CDS increases uncontrollably by approximately 60%. At that point is less prominent to find the best situated nodes in the network and therefore more nodes are selected to participate in the CDS. Focusing on the evaluation of the competitors, their performances follow the same pattern as in that of previous subsection. To elaborate, *EUCLB* still remains the champion algorithm but now is closely followed by *E2CLB* (or even loses by him e.g., in Layer 1 when diameter = 5 and when diameter = 70, or in Layer 3 when diameter = 70). In MCDS the size of the CDS is from 35% (best case) up to 92% (worst case) larger than that of *EUCLB*. The larger *per layer* differences in the CDS size are noted when diameter = 70. The reason for this is twofold. First, larger settings in the diameter parameter result in sparser vicinities in the network. Second, the sparser vicinities make the pruning process in *EMCDS* less efficient when only 2-hop neighborhood information is used. As about *E2WDB* it presents an almost equivalent performance with *E2CLB* when diameter = 5, 10 and 20 (up to 5% worse performance)and worse performance compared to *E2CLB* when diameter = 70 (from 10% up to 20%). Finally, the *E2MLB* CDS is up to 18% larger than that of *E2CLB*.

Fig. 7 illustrates the impact of network diameter on the *mean per layer node* minimum relay node energy. As expected, the first observation is that the generic trend is for *EUCLB* to present the worst performance among all the competitors. Interestingly, however it is even better than *EMCDS*⋆ when diameter = 70. This is due to the extremely larger CDS that *EMCDS*⋆ creates compared to *EUCLB* in conjunction with the sparser vicinities that exist in the network when diameter = 70. The best performing algorithm in this experiment is *E2CLB*. It presents comparable performance to *E2MLB* (approximately 5% better performance) when diameter = 70, which is getting even better for smaller settings of the diameter (up to 19% better performance when diameter = 5). Definitely, *E2CLB* can better distinguish between nodes that are situated relatively "close" to each other (smaller settings of the diameter),and select for the CDS those who have the larger residual energy. However, this positive performance gap diminishes in larger settings of the diameter, where sparser vicinities result to more nodes to be selected in the CDS.

On the other hand *E2MLB* is better than *E2WDB* (up to 20% better performance) and *EMCDS* (up to 20% better performance when diameter ≤ 40 and up to 31% better performance when diameter = 70). The worse performance of *EMCDS* is noted when diameter = 5 and when diameter = 70. In both cases, the root of the problem is the myopic look that *EMCDS* has that adds in the CDS many nodes with little energy in dense (diameter = 5) or sparse (diameter = 70) topologies compared to *E2WDB*. Concerning the impact of the mediator heuristic on the performance of *E2WDB* and *EM-*

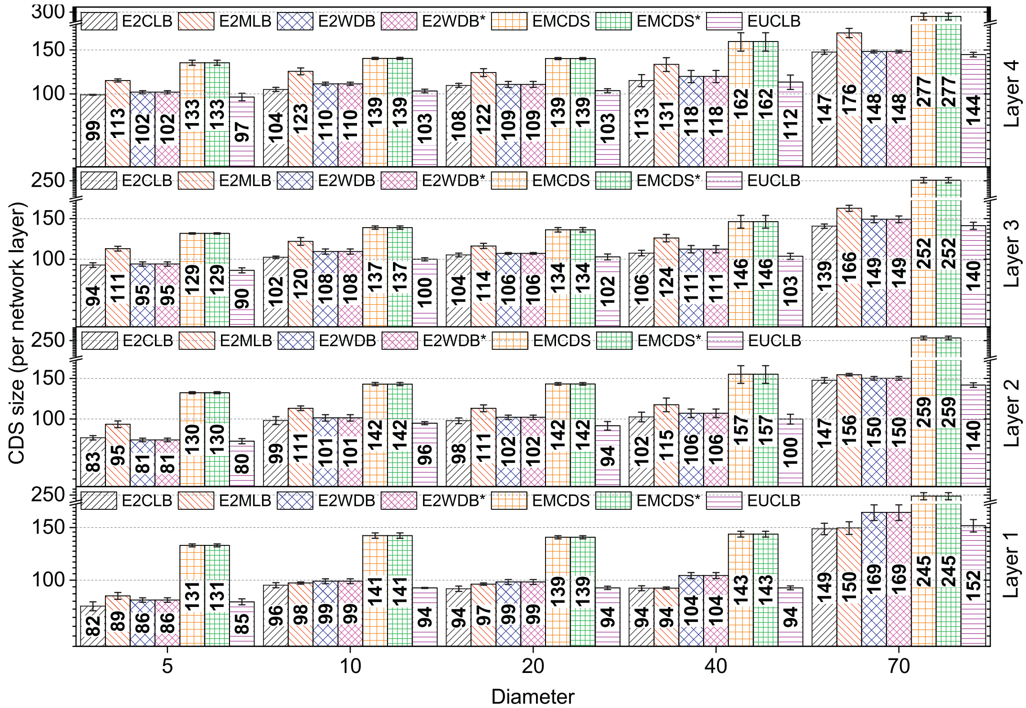

Fig. 6. Impact of network diameter on the size of CDS.

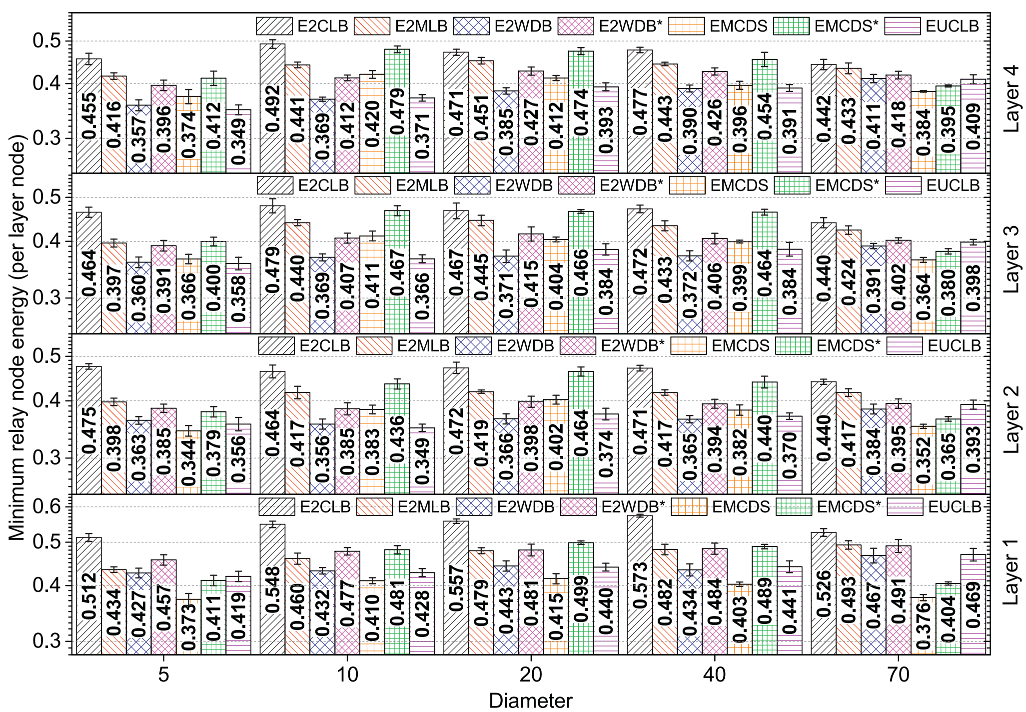

Fig. 7. Impact of network diameter on the energy level of each relay node (on the average the worst case scenario).

CDS, it is noteworthy that it improves the mean performance of *E2WDB*⋆ and *EMCDS*⋆ by 10% and 15% respectively. However, this performance improvement diminishes (it drops to approximately 5% for both cases) when the network topology is getting sparse (diameter = 70). Nevertheless, *E2MLB* presents better performance than *E2WD*⋆ (up to 8% better performance in all layers except for Layer 1 where *E2WD*∗ performance improves and gets up to 5% better than that of *E2MLB*) (Fig. 8).

Finally, in Fig. 9 as a brief statement of the most important information in a piece we summarize the aforementioned *per layer* results and present them in one diagram. From the bottom plot we conclude that generally the size of the CDS increases as the diameter parameter settings increase, for every algorithm. From the middle plot we conclude that generally the algorithms efficiency (in terms of the *mean per network node* minimum relay node energy) is considered irrespective to the network diameter. Finally, from the

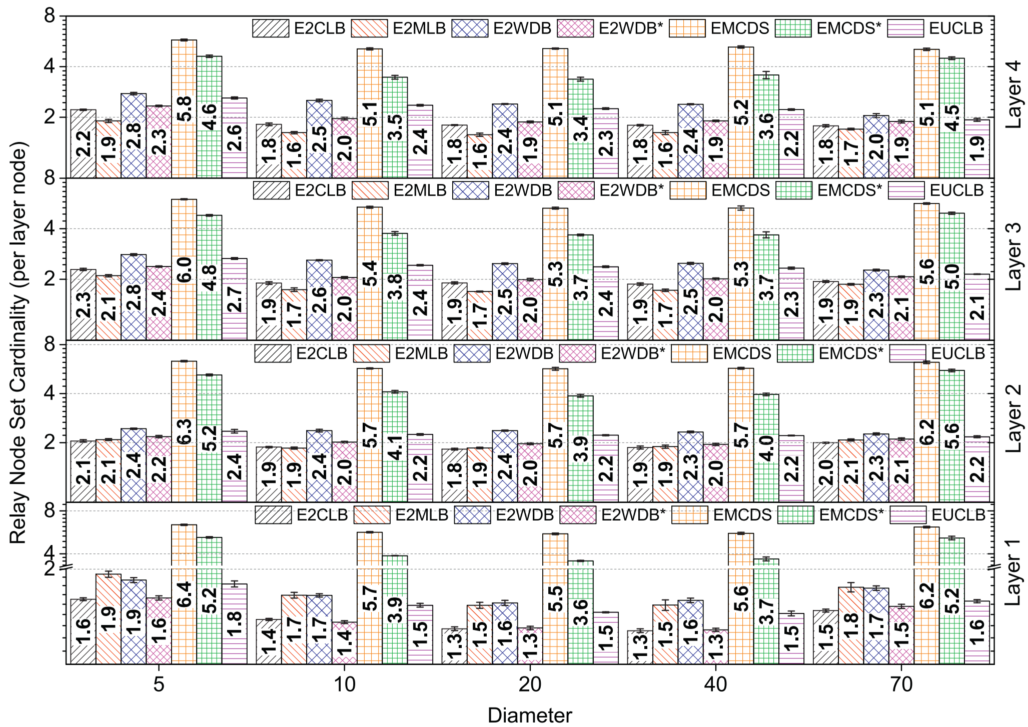

Fig. 8. Impact of network diameter on the size of the relay node set of each network node(on the average).

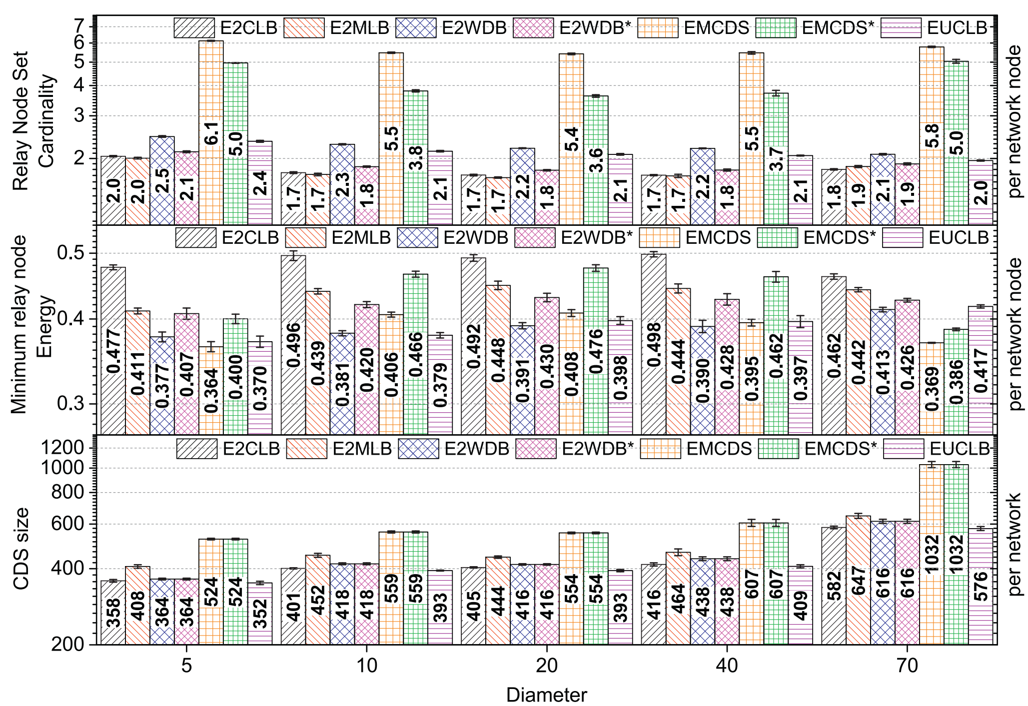

Fig. 9. Impact of network diameter on the performance of each algorithm.

upper plot we conclude that the *per network node* size of the relay node set is irrespective to the network diameter.

5.3.3. Impact of number of layers

In this section, we consider the impact of the number of network layers on the performance of each competitor. Firstly, in Fig. 10 we evaluate the *per layer* size of the CDS that each competitor creates. First, we note that the size of the CDS is a decreasing function with respect to the number of layers. This happens because as the number of layers increases it increases the number of interlinks among layers. Thus, the coverage capability of nodes that communicate with nodes in other layers increases which result to the reduced CDS. Focusing on the evaluation of the competitors, we observe that *EUCLB* remains the champion algorithm regarding the size of the CDS, followed by *E2CLB* (up to 10% worse performance), by *E2WDB* (up to 13% worse performance), by *E2MLB* (up to 29% worse performance) and finally by *EMCDS* (up to 71% worse performance).

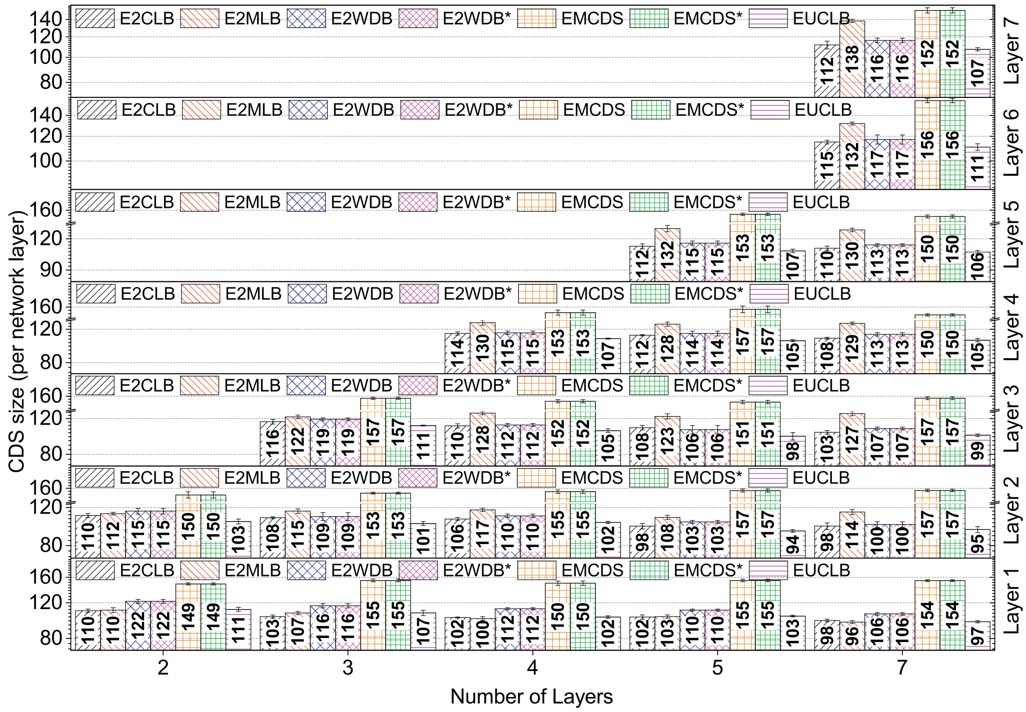

Fig. 10. Impact of the number of network layers on the size of CDS.

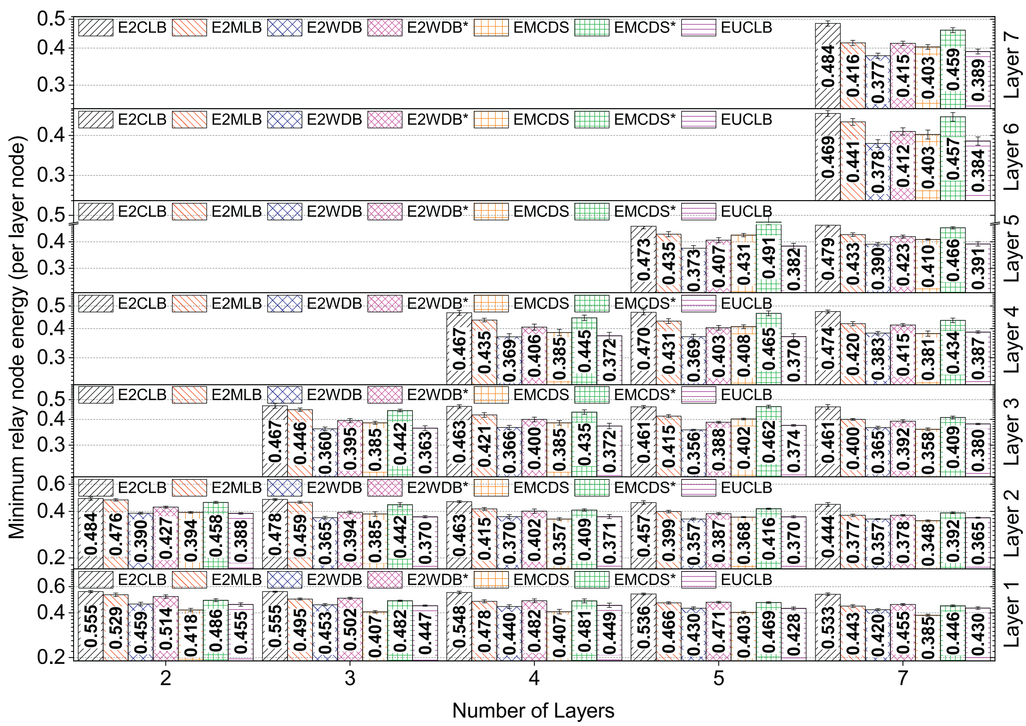

Fig. 11. Impact of the number of network layers on the energy level of each relay node (on the average the worst case scenario).

Fig. 11 illustrates the impact of the number of network layers on the *mean per layer node* minimum relay node energy. The first observation is that the *mean per layer node* minimum relay node energy is a decreasing function with respect to the number of layers (e.g., in layers 1, 2, 3). This is justified by the reduced *per layer* size of the CDS when considering an increasing number of network layers. As the size of the CDS is reduced it is reduced the likelihood of the less energy efficient nodes to be substituted by other more energy efficient nodes. The best performing algorithm is *E2CLB*, followed by *E2MLB* (up to 15% worse performance), next by *EUCLB* (up to 29% worse performance), next by *E2WDB* (up to 31% worse performance), and finally by *EMCDS* (up to 36% worse performance). The good performance of *EUCLB* compared to *E2WDB* and *EMCDS*, while it is unaware of the residual energy of the network nodes is due to the fact that energy-rich nodes are centrally situated in the network. Finally, note that the mediator heuristic

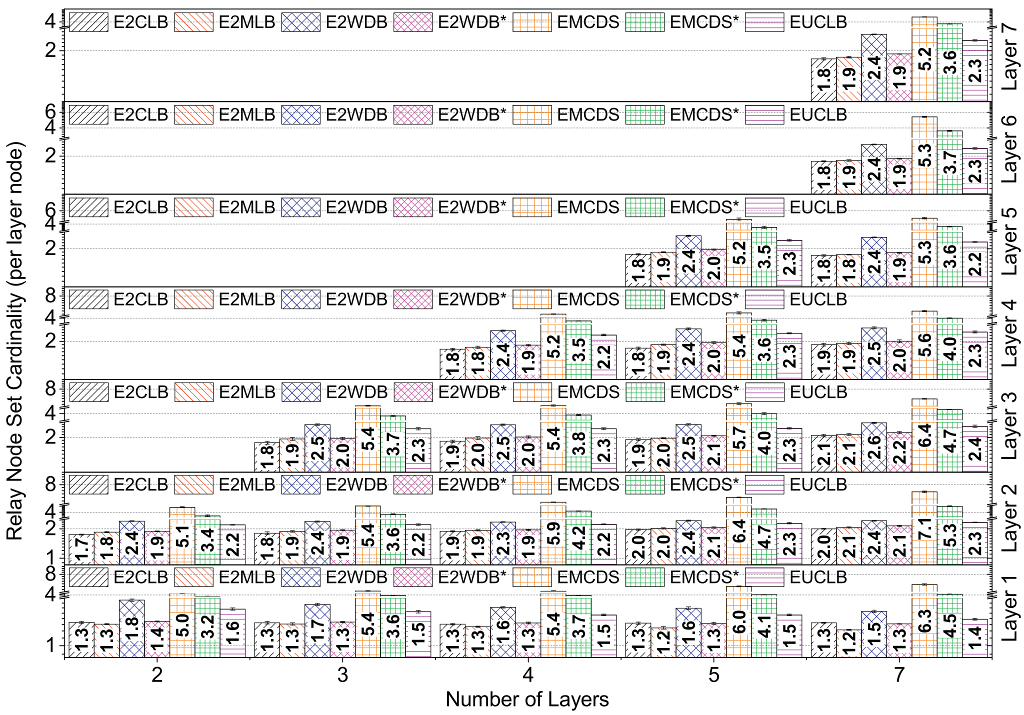

Fig. 12. Impact of the number of network layers on the size of the relay node set of each network node(on the average).

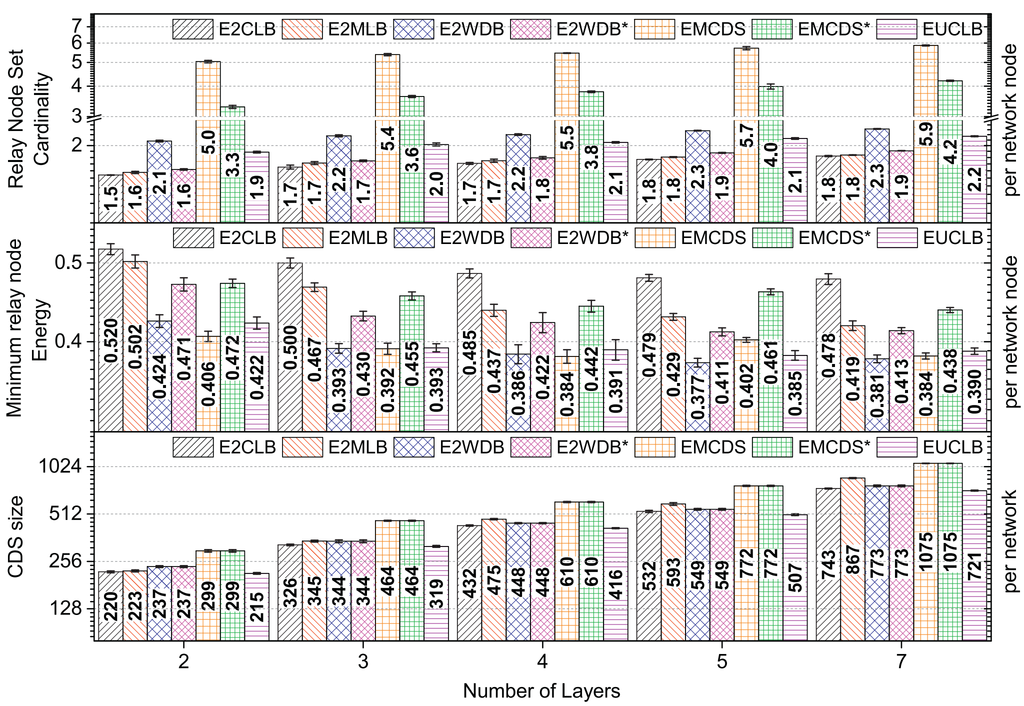

Fig. 13. Impact of the number of network layers on the performance of each algorithm.

improves the performance of *E2WDB* and *EMCDS* by up to 12% and 24%, respectively.

Next, in Fig. 12 we evaluate the *mean per layer node* relay set cardinality. The best performing algorithm is *E2CLB* and then follows *E2MLB, E2WDB*⋆, *E2WDB, EUCLB*. The mediator heuristic improves the efficiency of *E2WDB* and *EMCDS* regarding the *per layer node* relay node set cardinality (on the average) by 14% up to 31% for the *E2WD*∗ and by 34% up to 56% for the *EMCDS*⋆.

Finally, in Fig. 13 we summarize the aforementioned *per layer* results and present them in one diagram. From the bottom plot we conclude that the size of the CDS decreases as the number of layers increases, for every algorithm. It is straightforward that the larger the number of layers is the larger the need for more nodes to participate in the CDS becomes. From the middle plot, we conclude that generally the CDS efficiency (in terms of the *mean per network node* minimum relay node energy) is inversely proportional to the number of network layers. This is due to the se-

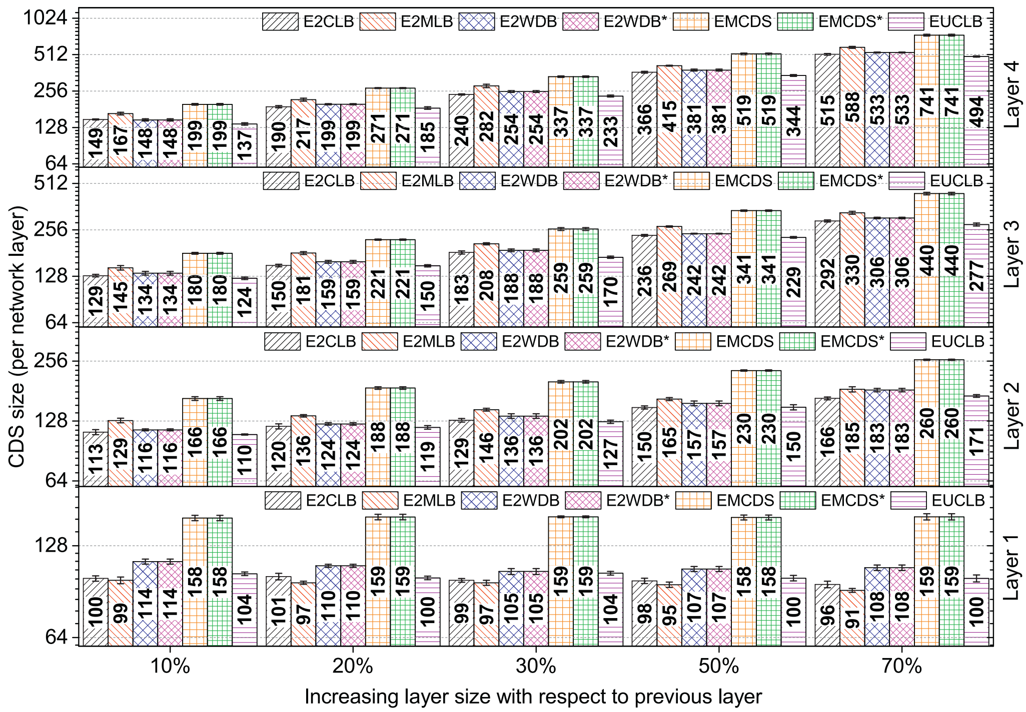

Fig. 14. Impact of the network size on the size of CDS.

lection of nodes for the CDS is driven primarily from the network topology, i.e., to establish network connectivity, and not from the energy level of each network node. The performance decrease approximately is 9% for *E2CLB*, 20% for *E2MLB*, 11% for *E2WDB* (14% for *E2WDB*⋆), 6% for *EMCDS* (8% for *EMCDS*⋆) and 8% for *EUCLB*. Finally, from the upper plot we conclude that the *per network node* size of the relay node set is irrespective to the number of network layers. Note that the mediator heuristic improves the efficiency of *E2WDB* and *EMCDS* regarding the *per network node* relay node set cardinality by 21% up to 31% for the *E2WD*⁎ and by 40% up to 52% for the *EMCDS*⋆.

5.3.4. Impact of increasing the layer size

In this section, we consider the impact of increasing the layer size on the performance of each competitor. Firstly, in Fig. 14 we evaluate the *per layer* size of the CDS that each competitor creates. Note that the size of the CDS is an increasing function with respect to the increasing layer size (except for Layer 1). This happens because as the size of each layer increases it increases the need for more nodes to act as connectors and thus for more nodes for the CDS. Focusing on the evaluation of the competitors, we observe that in the majority of cases *EUCLB* remains the champion algorithm regarding the size of the CDS, followed by *E2CLB* (up to 6% worse performance), by *E2WDB* (up to 8% worse performance), by *E2MLB* (up to 21% worse performance), and finally by *EMCDS* (up to 59% worse performance).

Fig. 15 illustrates the impact of increasing the layer size on the *mean per layer node* minimum relay node energy. The first observation is that the *mean per layer node* minimum relay node energy is irrespective to the increasing layer size. This is justified by the increased *per layer* size of the CDS when considering an increasing number of network layers. As the size of the CDS is increased it is more likely that the less energy efficient nodes to be substituted by other more energy efficient nodes. The best performing algorithm is *E2CLB*, followed by *E2MLB* (up to 18% worse performance), next by *EUCLB* (from 16% up to 25% worse performance), by *EMCDS* (from 12% up to 31% worse performance), and finally by *E2WDB* (from 19% up to 29% worse performance). Once again the weight distribution on the topology is responsible for the better performance of *EUCLB* compared to *E2WDB* and *EMCDS*, (energy efficient nodes are centrally situated in the network). Moreover, note that with the mediator heuristic the performance of *E2WDB*⋆ and *EMCDS*⋆ is improved compared to their "clean" versions by up to 10% and 18%, respectively.

Next, in Fig. 16 we evaluate the *mean per layer node* relay node set cardinality. In this experiment both *E2CLB* and *E2MLB* compete for presenting the best performance (without though having a clear winner), followed by *EUCLB* and *E2WDB*. Note that, the mediator heuristic improves the efficiency of *E2WDB*⋆ and *EMCDS*⋆ regarding the *per layer node* relay node set cardinality (on the average) by 5% up to 26% for the *E2WD*⁎ and by 11% up to 53% for the *EMCDS*⋆.

In Fig. 17 we summarize the aforementioned *per layer* results and present them in one diagram. From the bottom plot we conclude that the size of the CDS increases with increasing (with respect to the previous layer) layer size, for every algorithm. From the middle plot we conclude that generally the algorithms' efficiency (in terms of the *mean per network node* minimum relay node energy) is considered irrespective to the increasing layer size. Finally, from the upper plot we conclude that the *per network node* size of the relay node set is irrespective to the increasing layer size.

5.4. Evaluation of network load

In this section we evaluate the network load on nodes included in the CDS. In each experiment, we examined the simultaneous communication among distinct pairs of nodes (randomly selected) in a dozen of topologies with the same characteristics and measured the average queue length. Here, we include a small indicative subset of the obtained results. In particular, we show the results which concern the simultaneous communication between 200 pairs of nodes, and record the queue length of each CDS node. The overall conclusion is that all queues remain bounded,

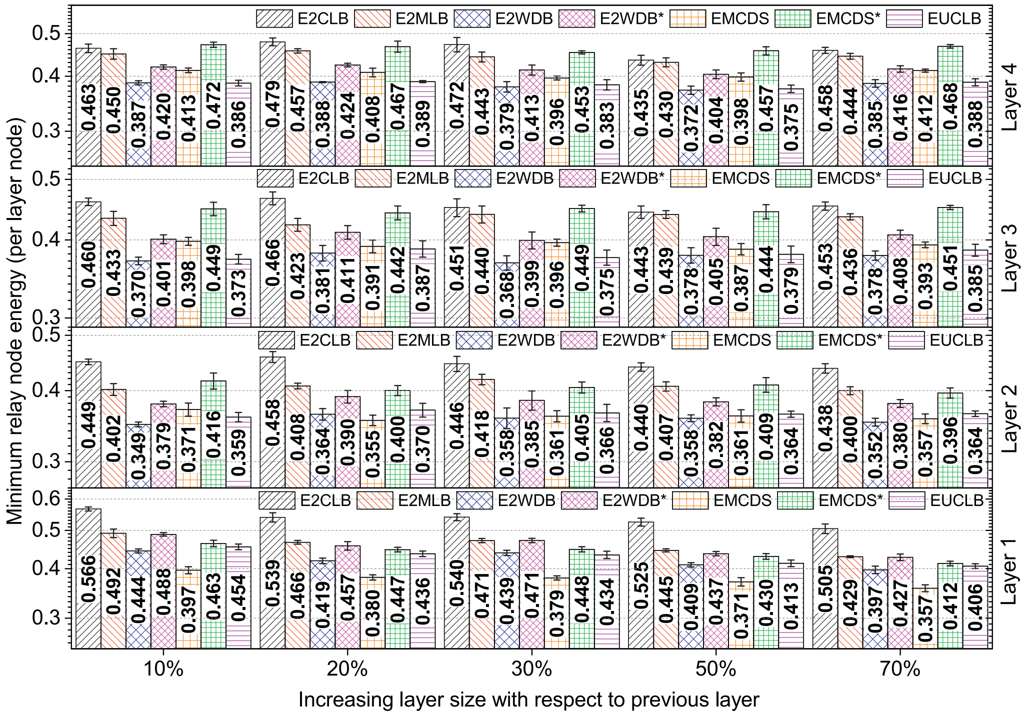

Fig. 15. Impact of the network size on the energy level of each relay node (on the average the worst case scenario).

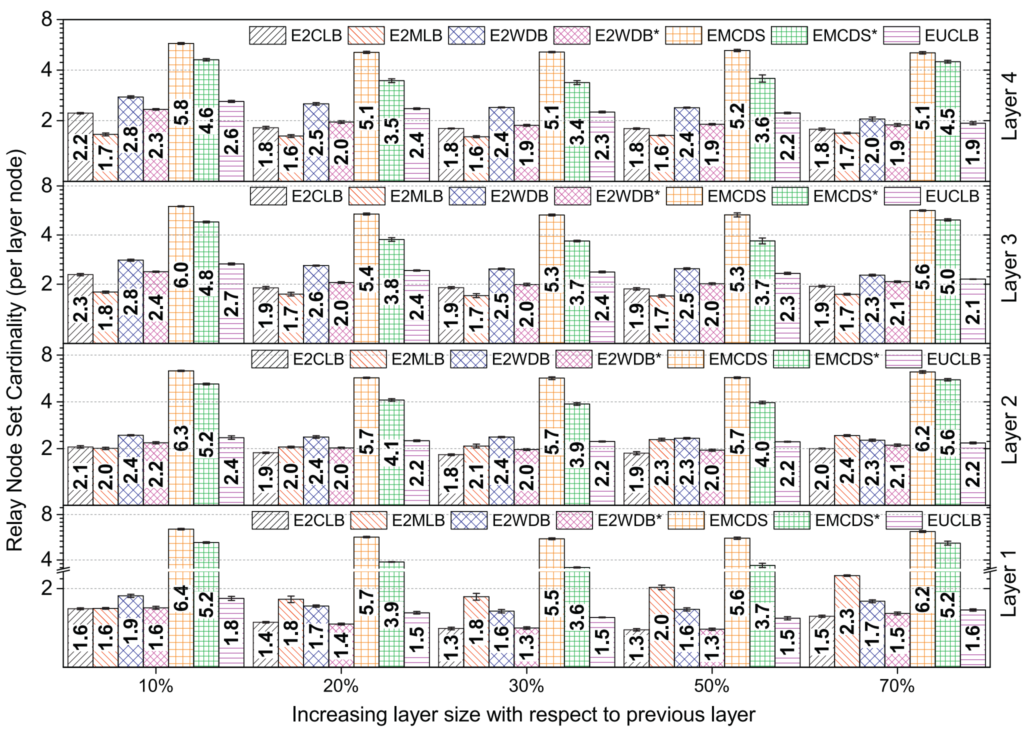

Fig. 16. Impact of the network size on the size of the relay node set of each network node (on the average).

and in particular only a couple of node queues reach a size of maximum eleven messages.

5.4.1. Network load in sparse networks

Fig. 18 illustrates the network load on the CDS nodes when considering sparsely connected multilayer networks. Namely, we utilized networks consisting of 4 equi-sized layers, with a total number of nodes equal to 2000 and layer diameter equal to 70. The generic observation is that none of the competing methods presents any likely-overflow buffer phenomenon. Moreover, the *E2CLB* algorithm manages to have the largest number of nodes with the least number of messages, i.e., around 800 nodes whose queue accommodates on the average one message.

5.4.2. Network load in dense networks

Next, in Fig. 19 we examine the network load on dense multilayer networks. The setting is same as previously, but now with a diameter of each layer equal to around 50. We observe the same pattern of performance as in the previous experiment, and we see – for all competitors – fewer nodes with larger queues which is to

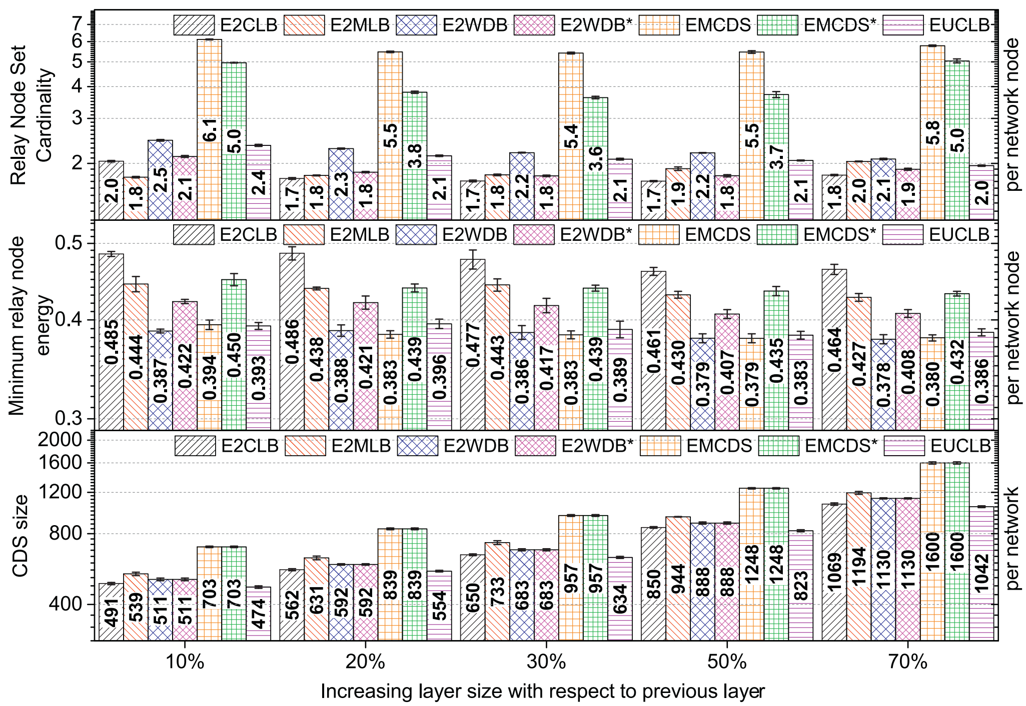

Fig. 17. Impact of increasing the layer size on the performance of each algorithm.

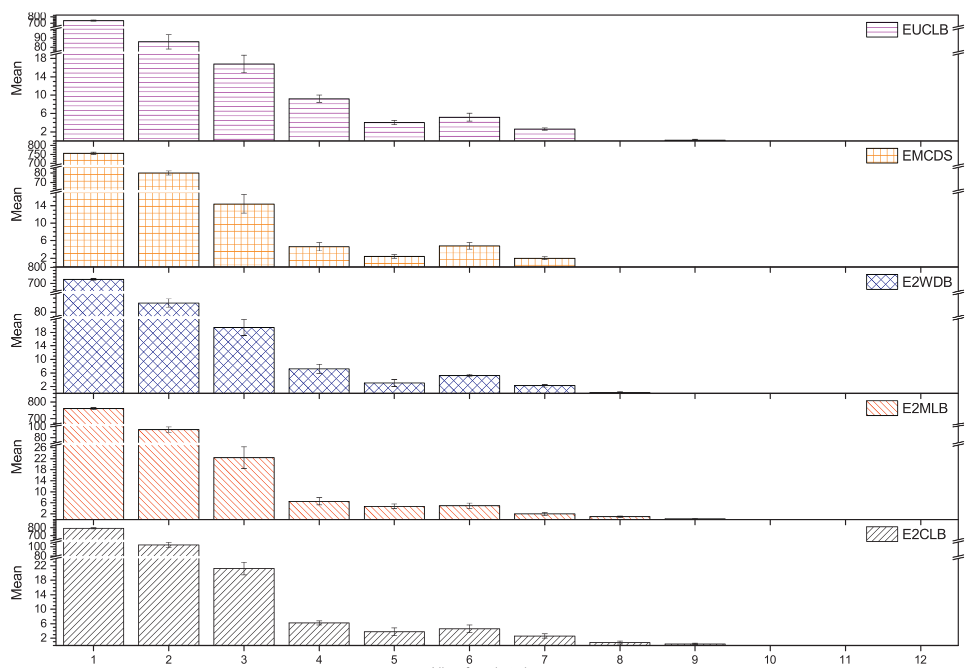

Fig. 18. Network load in sparse networks.

be expected since in dense networks more paths are available to serve traffic.

5.4.3. Network load in networks with more layers

Next, in Fig. 20 we examine the load on networks with more layers, namely with 7 equi-sized layers (now the number of nodes is 3500). The results are alike the previous experiment, since now the communicating pairs are spread more sparsely among the set of nodes.

5.4.4. Network load in networks with non equi-sized layers

Finally, in Fig. 21 we examine the network load on when the layers differ in their size. The setting is as the original one, but we have networks with 4 layers, and the adjacent layers differ in the number of nodes by 20%, (the total number of nodes is 2184). The difference in queue lengths is that we observe an increase in the number of nodes with moderate queue size, because some CDS nodes which belong to layers with less nodes are selected for message routing.

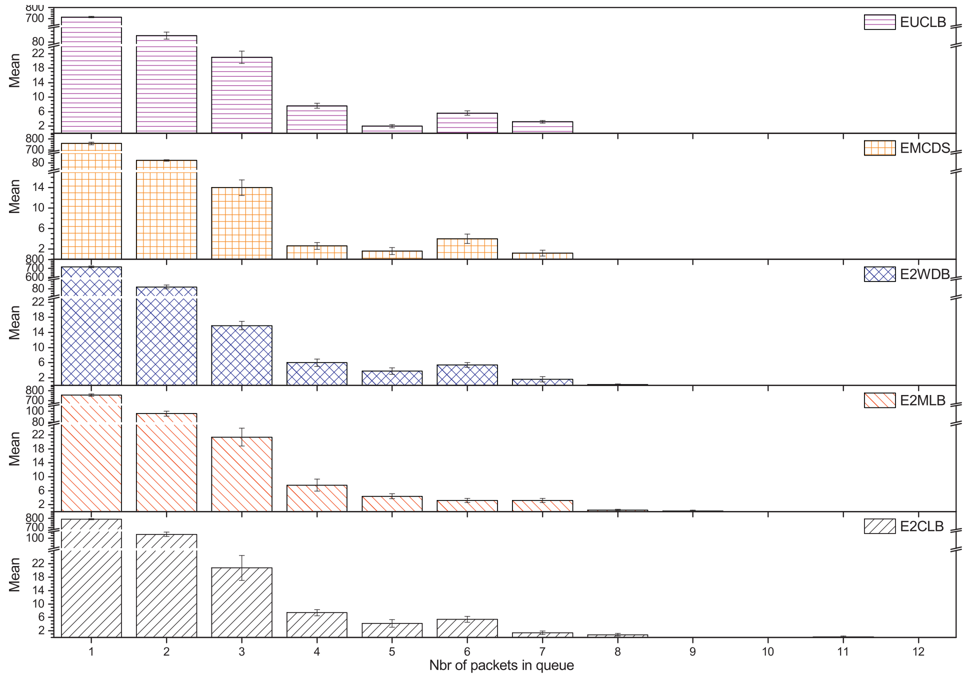

Fig. 19. Network load in dense networks.

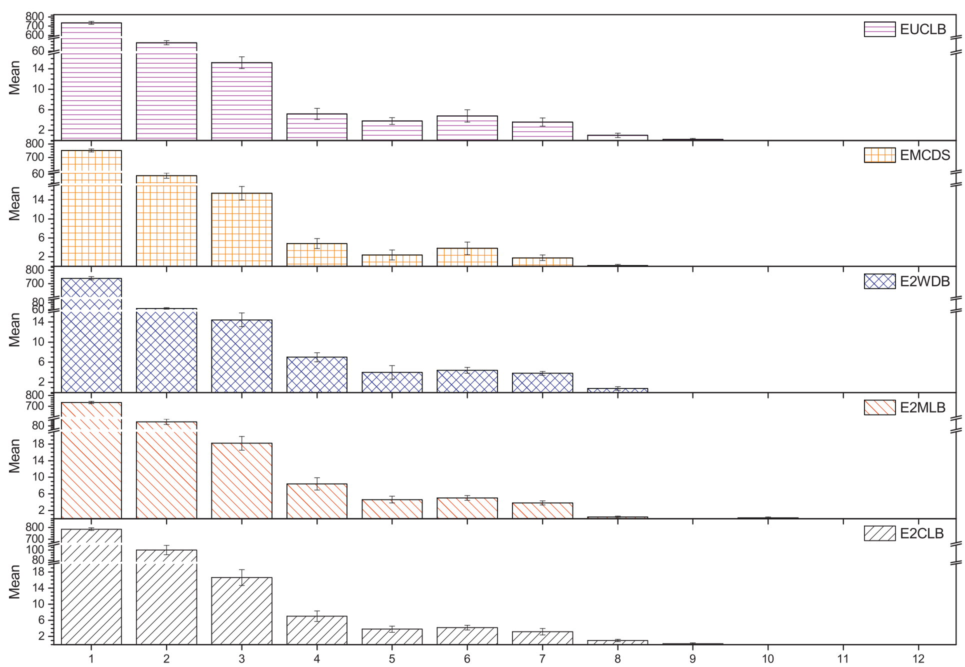

Fig. 20. Network load in networks with more layers.

6. Related work

The topic of backbone construction algorithms for wireless ad hoc networks has nearly a two decades long history, and it mainly features methods categorized as either cluster-based [31] or dominating set-based [7,8]. Protocols belonging to the former category are facing adoption difficulties due to mobility or assume predefined mobility [6,32]. The latter category algorithms can combine flexibility – by optimizing for backbone diameter, asymmetry in transmission range, interference – with social-cognitive techniques [11]. Energy conservation is of paramount significance in many implementation of ad hoc networks because several of the participating entities are energy starving devices. There is also a lot of work on developing approaches for energy conservation in protocols for wireless ad hoc networks [33].

Centrality concepts have been exploited widely in ad hoc networking for purposes of cooperative caching [11], service deployments [34], access control [35], security [36], routing [37], in many areas of delay tolerant networking [38], and so on.

Multilayer networks [39] are a particularly hot research area of network science. In [13] we developed several centrality measures for helping in the identification of influential spreaders [40] in

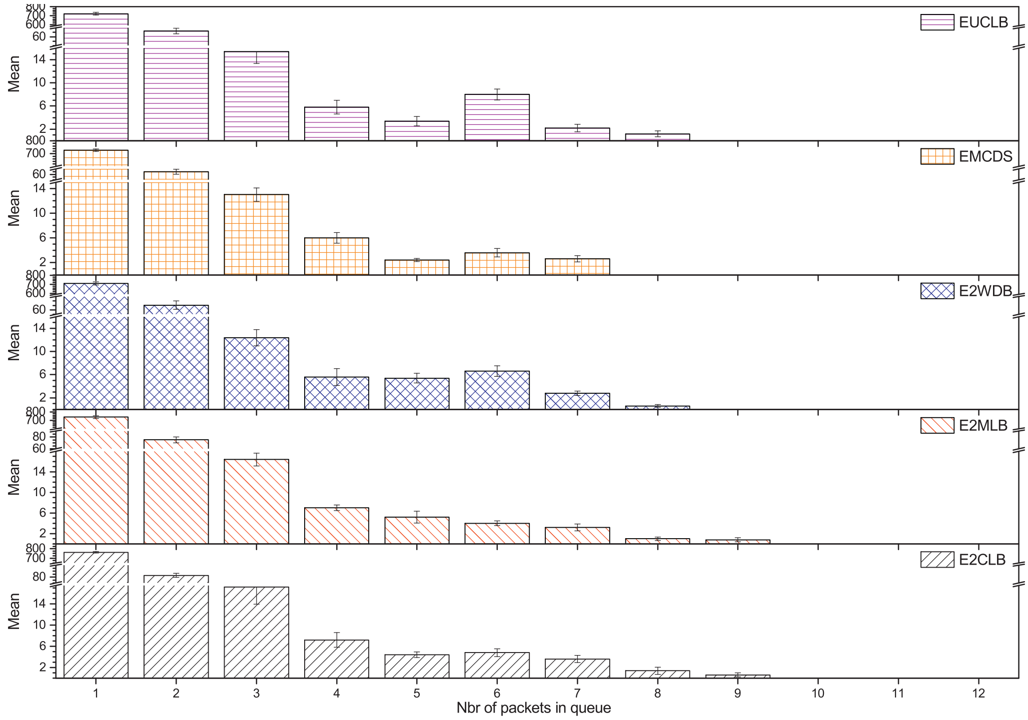

Fig. 21. Network load in networks with unequal layer sizes.

multilayer networks. Multilayer network literature has not been exploited widely yet in the area of ad hoc networking even though several of its advances can be applied there. Calculating connected dominating sets with the purpose of operating as backbones in wireless multilayer ad hoc networks was studied so far only in [3]; it was proved there that some peculiarities of the problem make existing solutions either not appropriate or not efficient. However, that work did not consider energy conservation issues that the present work focuses on.

7. Conclusion

Multilayer wireless ad hoc networks arise in several modern settings and pose some new challenges around effective and efficient communication capability among their entities. This article investigates the problem of constructing energy-aware backbones for such network types utilizing the notion of connected dominating sets (CDS). Improving on from our earlier work, which proved the insufficiency of traditional algorithms for addressing this problem, we proposed a new centrality measure, namely *EclPCI* which can identify energy-rich and at the same time "central" to the topology nodes. Then, the article developed a distributed algorithm, namely *E2CLB* for calculating an energy-aware connected dominating set based on the proposed centrality measure.

The proposed algorithm was evaluated analytically by establishing its computational and communication complexity, and experimentally in an exhaustive manner. The experimental evaluation was done with respect to independent parameters that quantify the structure of the topology, i.e., density and shape (diameter), the size of the multilayer network in terms of the number of nodes and layer. The performance measures quantified the overall (and per layer) size of the dominating set, and the residual energy. Even though there is no prior related work on this subject, we employ as competitors six other algorithms; some of them stem from the present work and others are straightforward extensions of traditional well-known algorithms. In all experiments, the proposed *E2CLB* proved to be the winning algorithm in the sense that it could trade a very small increase or no increase at all at the dominating set size in order to offer significant gains in terms of residual energy of the CDS nodes. Interesting extensions of the present work are the investigation of this problem for unidirectional connectivity or the incremental maintenance of the CDS in cases of topology changes.

Acknowledgments

This work was partially supported by the US Office of Naval Research (ONR) under award N00014-14-1-2190. D. Katsaros acknowledges the support of a Yale grant for conducting research on "Data Analytics for Multi-Layer Networks".

Appendix A. Results with skewness-varying topologies

Here we evaluate the efficiency of the competing algorithms when considering networks whose topologies and weight skewness settings varying across a range of settings. Each multilayer network is composed by 4 layers, each one of them containing 500 nodes (mean degree = 6). In Table 2 we present for the three different settings of the topology skewness (Low, Medium, and High) the values of the respective parameters of interest. In Figs. 22–24 the results concern the case where the skewness is towards high degree nodes, and in Figs. 25–27 the results concern the case where the skewness is towards low degree nodes. The generic observation is that the performance differences between competitors remain almost stable regardless of the topology and weight skew-

Table 2
Parameter values.

Topology Skewness	s_{degree}	s_{layer}	s_{node}
Low	0.1	0.1	0.1
Medium	0.5	0.5	0.5
High	0.9	0.9	0.9

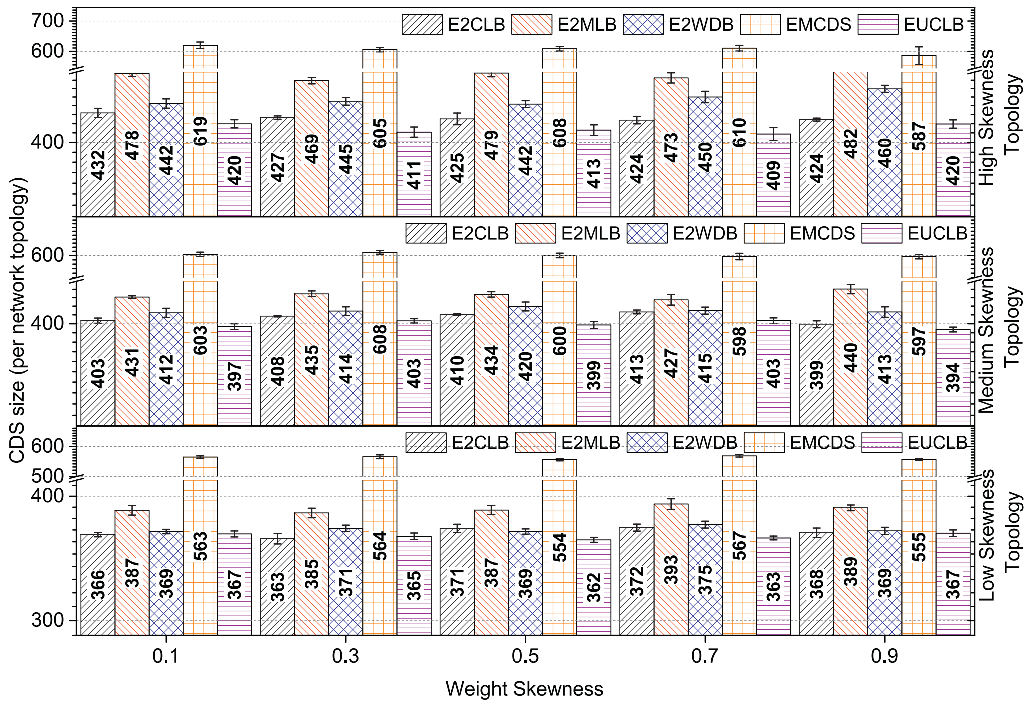

Fig. 22. Algorithms performance (CDS size) with skewness to high degree nodes.

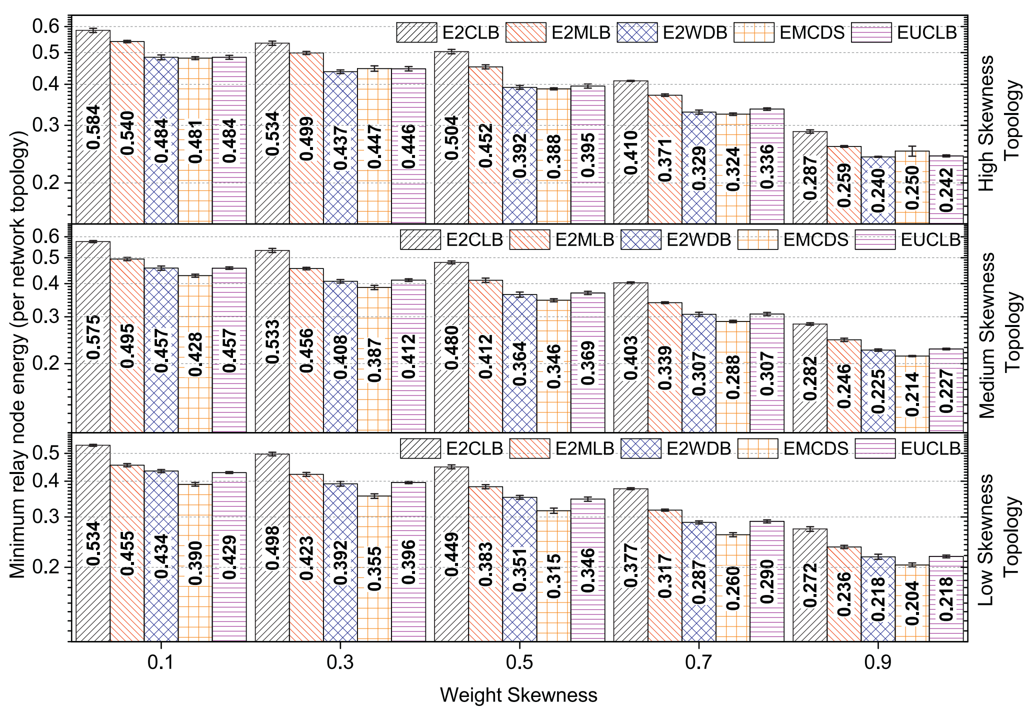

Fig. 23. Algorithms performance (min energy) with skewness to high degree nodes.

ness. This justifies the fairness of the settings we used in our earlier experimentation.

A1. Skewness to high degree nodes

In Fig. 22 we evaluate the impact of the various settings of the topology and weight skewness on the performance of the competing algorithms regarding the size of the CDS when skewness is towards high degree nodes. The first observation is that the size of the CDS increases for larger settings of the topology skewness with respect to the same weight skewness setting. That is something we expected to happen as larger settings of the topology skewness result to non-uniform distribution of the interlinks among the mlNetwork layers, the appearance of some hub nodes in the mlNetwork and consequently drives to more *per layer* nodes selected for the CDS in order to guarantee the network connectivity.

The second observation concerns the impact of the weight skewness on the size of the CDS with respect to the same topology skewness settings and should be considered in combination

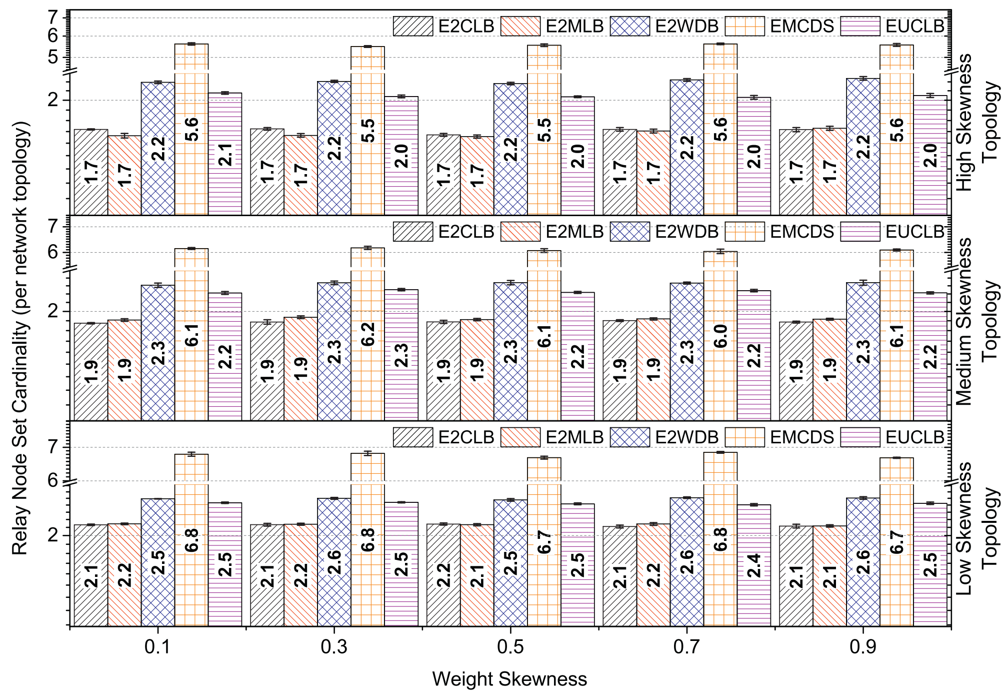

Fig. 24. Algorithms performance (relay node set cardinality) with skewness to high degree nodes.

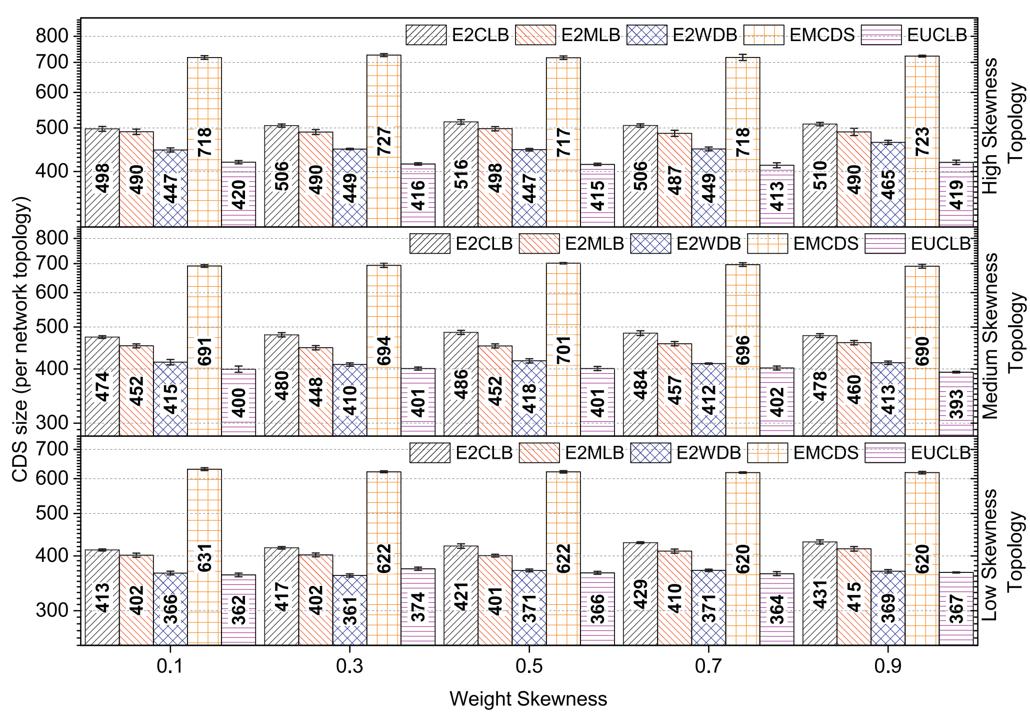

Fig. 25. Algorithms performance (CDS size) with skewness to low degree nodes.

with the respective results of Fig. 23. To elaborate, note that the weight skewness has negligible impact on the size of the CDS for the same topology skewness settings. That is happening because the algorithms decide about the CDS primarily based on the existing network topology (establish network connectivity). The residual energy is taken into account only when there is some coverage redundancy between the mlNetwork nodes(establish network connectivity first and then strive to substitute the less energy efficient nodes). This observation justifies the case in Fig. 23 where the *mean per network node* minimum relay node energy decreases for larger settings of the weight skewness as larger settings of the weight skewness result to less uniform distribution of the weights in the mlNetwork and consequently to the selection of some less energy efficient nodes in the CDS.

Finally, in Fig. 24 we observe that the weight skewness has negligible impact on the *mean per network node* size of the relay node set which is justified by the fact that each algorithm strives for the minimum possible *per network node* relay node set as this guarantees smaller volume of broadcast message transmissions in the network. We observe also that for larger settings of the topology

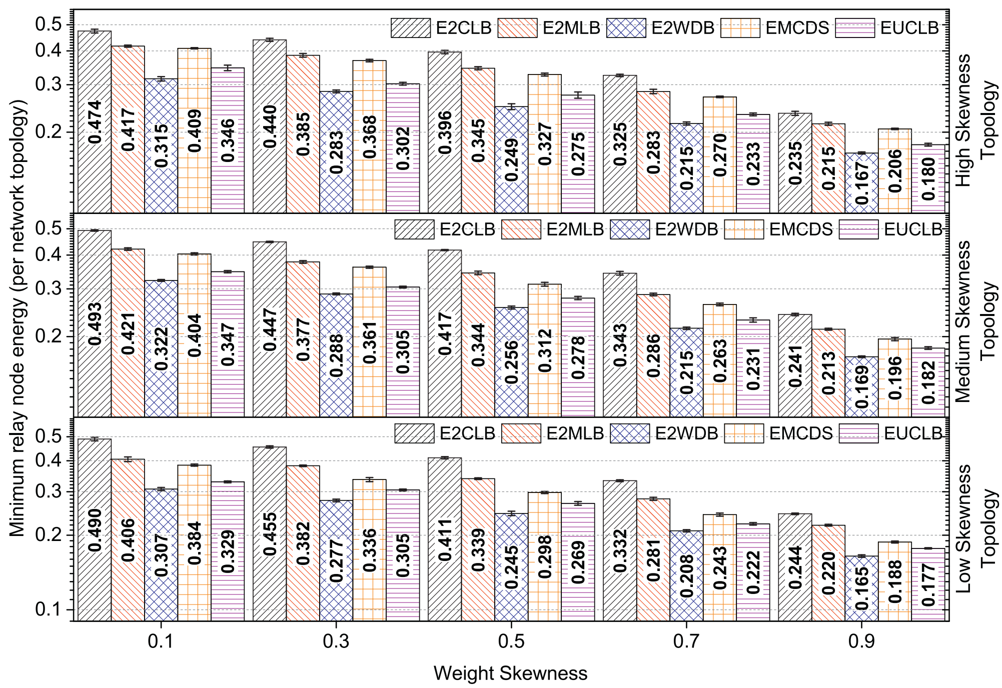

Fig. 26. Algorithms performance (min energy) with skewness to low degree nodes.

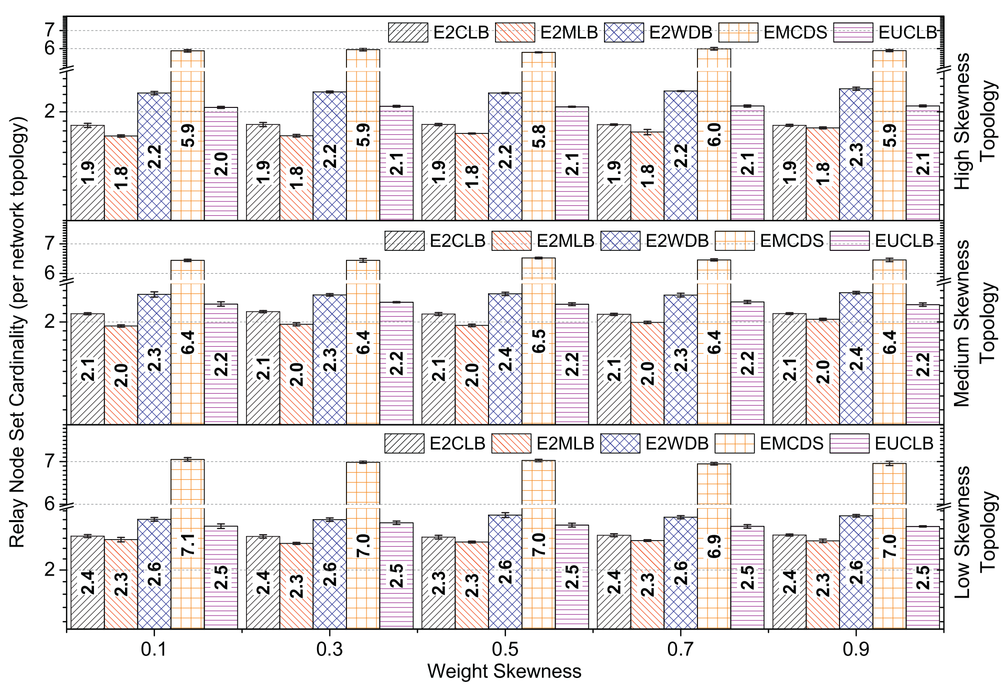

Fig. 27. Algorithms performance (relay node set cardinality) with skewness to Low degree nodes.

skewness the *mean per network node* cardinality of the relay node set decreases which is justified by the larger CDS with these settings and thus the greater coverage capability of the selected relay nodes.

A2. Skewness to low degree nodes

In Fig. 25 we evaluate the impact of the various settings of the topology and weight skewness on the performance of the competing algorithms regarding the CDS when the skewness is towards low degree nodes. The observations of Fig. 22 regarding the size of the CDS when using larger settings of the topology skewness still apply. Nevertheless, the performance of both *E2CLB* and *E2MLB* worsens compared to the respective performance of *EUCLB* and *E2WDB*. That is because of the attributes of the mlNetwork; i.e. low degree nodes take priority over high degree nodes in getting the interlinks which makes them good choices for the CDS. However, the reduced coverage of the low degree nodes in combination with the residual energy of the participating nodes in the CDS which we should take into consideration explains the larger CDS of *EUCLB* and *E2WDB*. All in all, that is acceptable to happen as long as both *E2CLB* and *E2MLB* create energy efficient CDSs.

In Fig. 26 we evaluate the impact of the various settings of the topology and weight skewness on the performance of the competing algorithms regarding the *mean per network node* minimum relay node energy when the skewness is towards low degree nodes. The observations of Fig. 23 still apply; i.e. the *mean per network node* minimum relay node energy decreases for larger settings of the weight skewness. Moreover, the relative performance among competing algorithms compared to when the skewness is towards high degree nodes still apply except for *E2WDB* which presents worse performance by *EMCDS*.

Finally, in Fig. 27 we observe that the weight skewness has negligible impact on the *mean per network node* size of the relay node set. We observe also that for larger settings of the topology skewness the *mean per network node* cardinality of the relay node set decreases which is justified by the larger CDS with these settings; i.e. the larger CDS is a by product of larger relay node sets which results in increased likelihood that a less efficient relay node to be substituted by an energy efficient relay node.

Appendix B. Pruning rule *k* efficiency

In this section we evaluate the efficiency on using more connectivity information in reducing the size of CDS during the pruning phase.

Impact of topology density. The results presented in Fig. 28 study the impact of increasing node degree on the performance measures when using 2-hop information, and Fig. 29 when using 3-hop information for all algorithms but *EMCDS*. The results are intuitive and confirm the findings of the main article. Denser connectivity (higher average degree) means smaller CDS, equal or larger relay node sets per node. Utilizing more information, i.e., 3-hop information can decrease these quantities by a factor of 2 or 3. Champions algorithms are as before.

Impact of network diameter. The results shown in Fig. 30 investigate the impact of increasing diameter on the performance of the competitors when exploiting 2-hop information or 3-hop information (Fig. 31). The performance patterns are similar to those reported in the previous pair of graphs. Using such rich information every algorithm can improve its performance concerning CDS size 3 times from small and medium diameter values, and 2 times for larger diameter values.

Impact of number of layers. The results shown in Fig. 32 investigate the impact of increasing the number of network layers on the performance of the competitors when exploiting 2-hop information or 3-hop information (Fig. 33). Here the performance gains are smaller and every competitor improves itself at a factor of 2 concerning CDS size, and at a factor of 1.5 concerning relay set size and residual energy.

Impact of increasing layer size. The results shown in Fig. 34 investigate the impact of increasing the number of network layers on the performance of the competitors when exploiting 2-hop information or 3-hop information (Fig. 35). The results are alike those observed in the previous pair of plots.

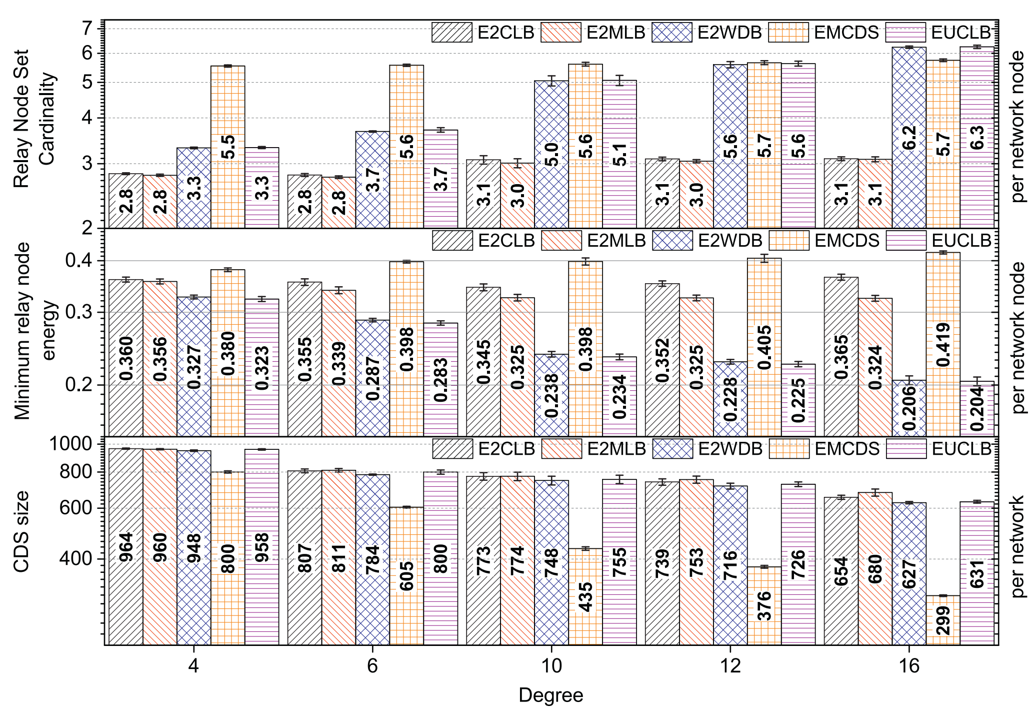

Fig. 28. Impact of network density on the performance of each algorithm by using 2-hop neighborhood information.

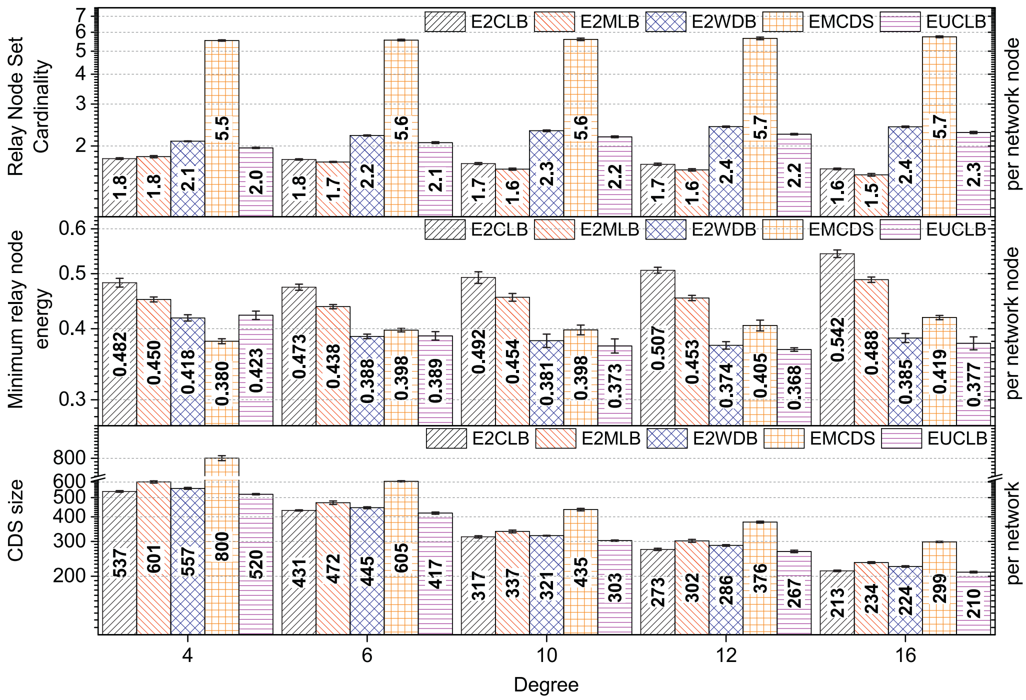

Fig. 29. Impact of network density on the performance of each algorithm when using 2-hop neighborhood information for *EMCDS*and 3-hop neighborhood information for the rest.

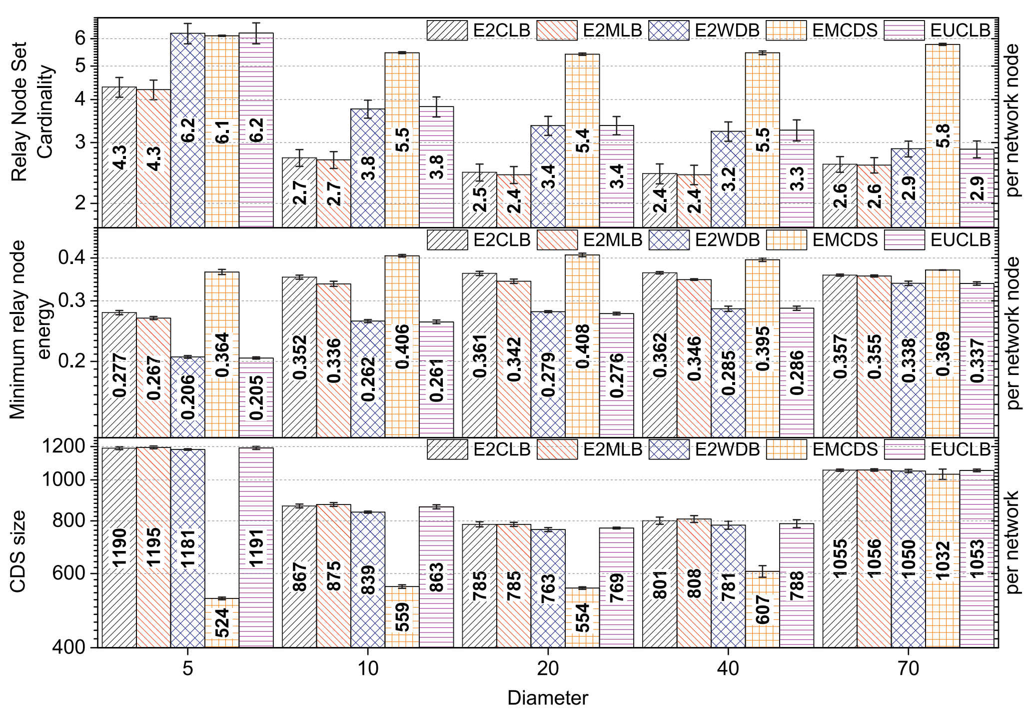

Fig. 30. Impact of network diameter on the performance of each algorithm when using 2-hop neighborhood information.

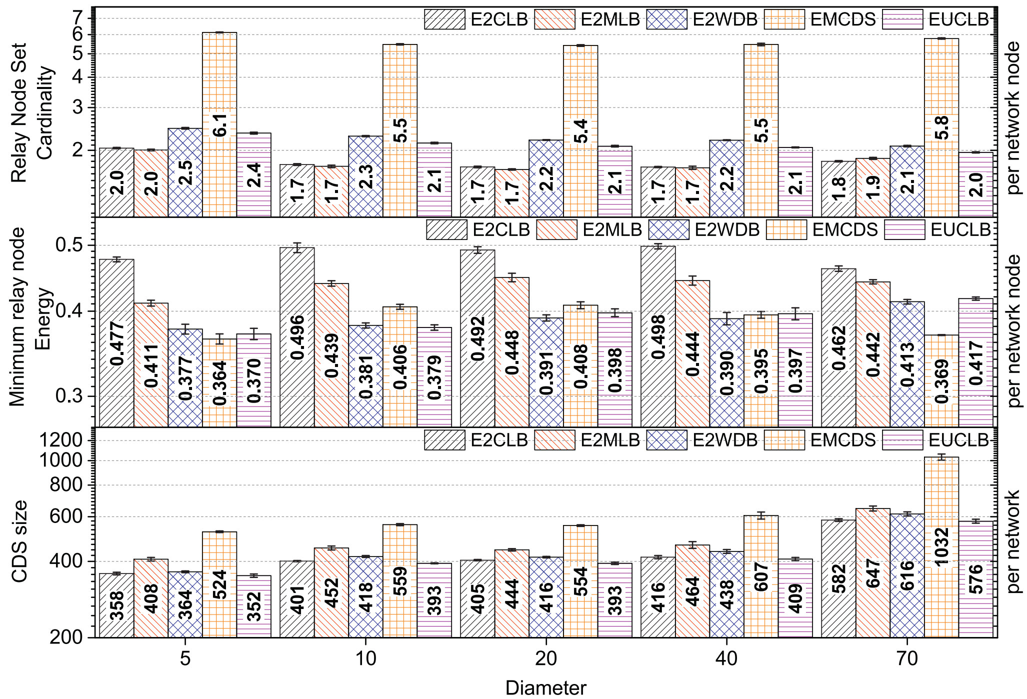

Fig. 31. Impact of network diameter on the performance of each algorithm when using 2-hop neighborhood information for *EMCDS*and 3-hop neighborhood information for the rest.

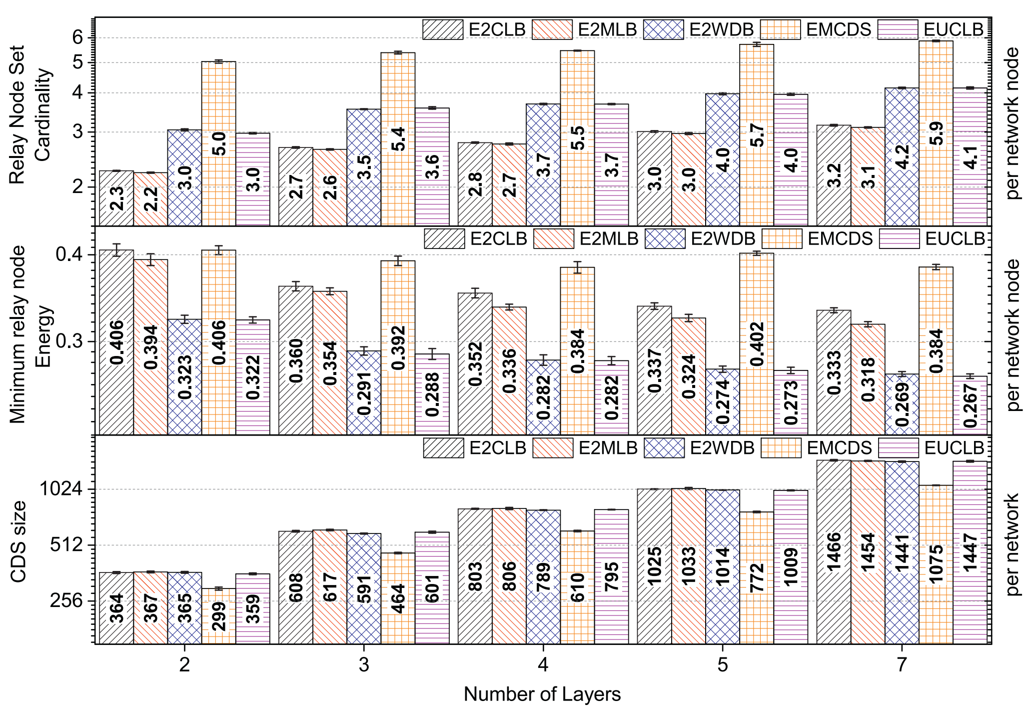

Fig. 32. Impact of the number of network layers on the performance of each algorithm when using 2-hop neighborhood information.

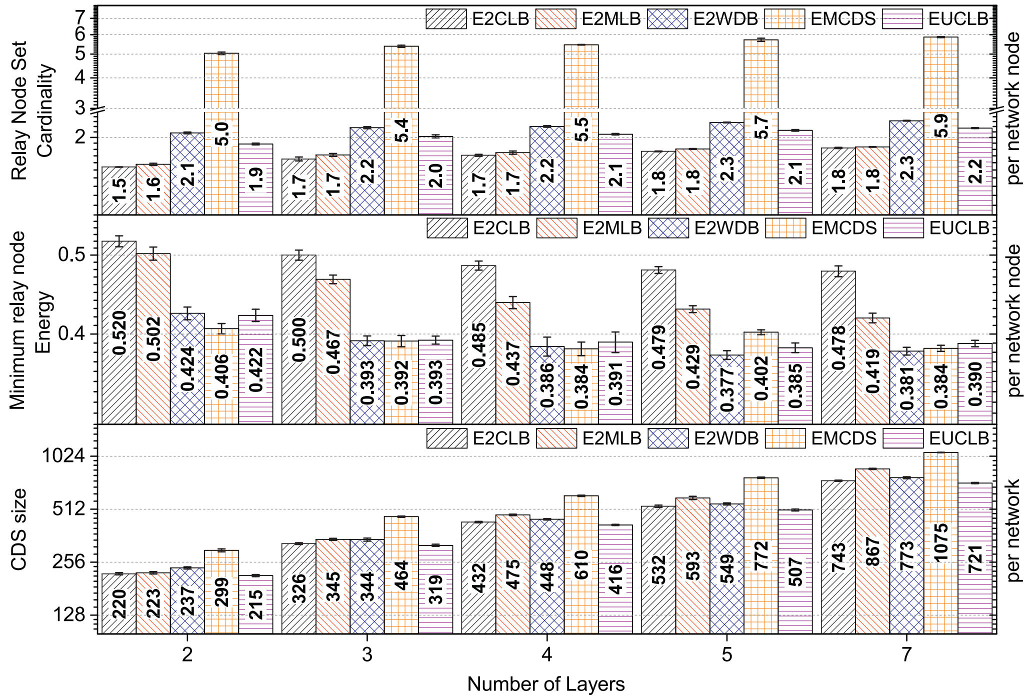

Fig. 33. Impact of the number of network layers on the performance of each algorithm when using 2-hop neighborhood information for *EMCDS* and 3-hop neighborhood information for the rest.

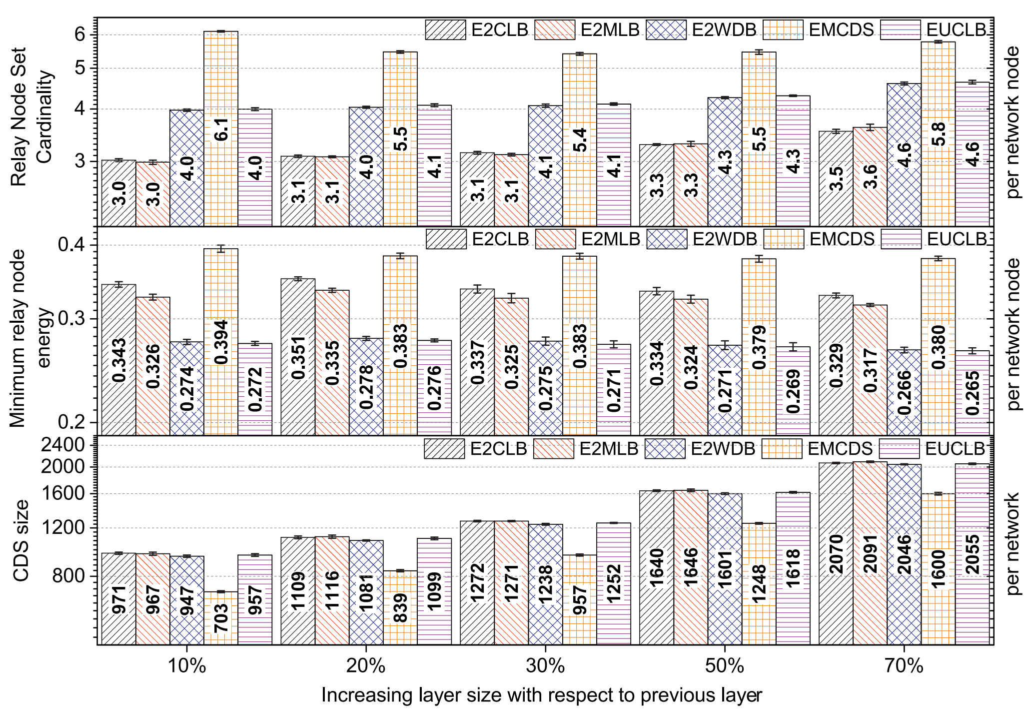

Fig. 34. Impact of increasing the layer size on the performance of each algorithm when using 2-hop neighborhood information.

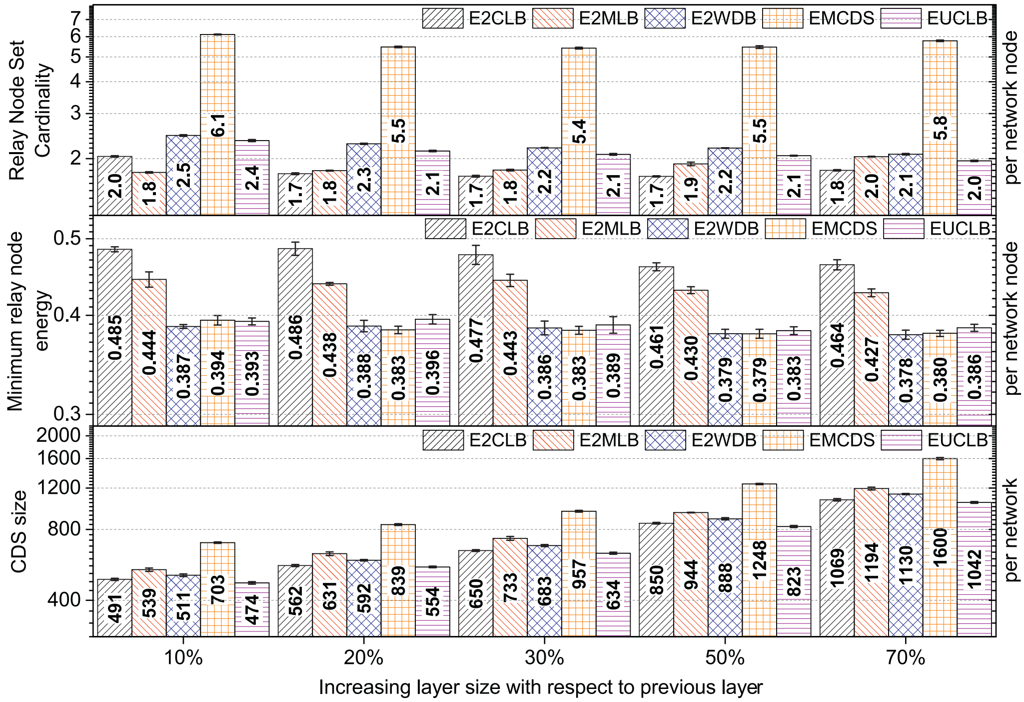

Fig. 35. Impact of increasing the layer size on the performance of each algorithm when using 2-hop neighborhood information for *EMCDS* and 3-hop neighborhood information for the rest.

References

[1] C.E. Fossa, T.G. Macdonald, Internetworking tactical MANETs, in: Proceedings of the IEEE Military Communications Conference (MILCOM), 2010.

[2] A.A. Gohari, R. Pakbaz, P.M. Melliar-Smith, L.E. Moser, V. Rodoplu, RMR: reliability map routing for tactical mobile ad hoc networks, IEEE J. Sel. Areas Commun. 29 (10) (2011) 1935–1947.

[3] D. Papakostas, P. Basaras, D. Katsaros, L. Tassiulas, Backbone formation in military multi-layer ad hoc networks using complex network concepts, in: Proceedings of the IEEE Military Communications Conference (MILCOM), 2016, pp. 842–848.

[4] D. Papakostas, D. Katsaros, A simulation-based performance evaluation of a randomized MIS-based clustering algorithm for ad hoc networks, Simul. Model. Pract. Theory 48 (2014) 1–23.

[5] O. Younis, M. Krunz, S. Ramasubramanian, Node clustering in wireless sensor networks: recent developments and deployment challenges, IEEE Netw. Mag. 20 (3) (2006) 20–25.

[6] L. Maglaras, D. Katsaros, Social clustering of vehicles based on semi-Markov processes, IEEE Trans. Veh. Technol. 65 (1) (2016) 318–332.

[7] P. Ruiz, P. Bouvry, Survey on broadcast algorithms for mobile ad hoc networks, ACM Comput. Surv. 48 (1) (2015) 1–35.

[8] J. Yu, N. Wang, G. Wang, D. Yu, Connected dominating sets in wireless ad hoc and sensor networks – a comprehensive survey, Comput. Commun. 36 (2) (2013) 121–134.

[9] M. Gerla, J.-C. Tsai, Multicluster, mobile, multimedia radio network, 1, 1995, pp. 255–265.

[10] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-completeness, W. H. Freenan and Company, 1979.

[11] N. Dimokas, D. Katsaros, Y. Manolopoulos, High performance, low complexity cooperative caching for wireless sensor networks, Wirel. Netw. 17 (3) (2011) 717–737.

[12] J.E. Hirsch, An index to quantify an individual's scientific research output, Proce. National Acad. Sci. 102 (46) (2005) 16569–16572.

[13] P. Basaras, G. Iosifidis, D. Katsaros, L. Tassiulas, Identifying influential spreaders in complex multilayer networks: a centrality perspective, IEEE Trans. Netw. Sci. Eng. (2018). Accepted for publication

[14] P. Thulasiraman, K.A. White, Topology control of tactical wireless sensor networks using energy efficient zone routing, Digital Commun. Netw. 2 (1) (2016) 1–14.

[15] M. Younis, I.F. Senturk, K. Akkaya, S. Lee, F. Senel, Topology management techniques for tolerating node failures in wireless sensor networks: a survey, Comput. Netw. 58 (2014) 254–283.

[16] M. Eidsaa, E. Almaas, *s*-core network decomposition: a generalization of *k*-core analysis to weighted networks, Phys. Rev. E 88 (062819) (2013) 062819–1–062819–9.

[17] N. Dimokas, D. Katsaros, Y. Manolopoulos, Energy-efficient distributed clustering in wireless sensor networks, J. Parallel Distrib. Comput. 70 (4) (2010) 371–383.

[18] A. Qayyum, L. Viennot, A. Laouiti, Multipoint relaying for flooding broadcast messages in mobile wireless networks, in: Proceedings of the Hawaii International Conference on System Sciences (HICSS), 2002, pp. 3866–3875.

[19] R. Song, J.D. Brown, P.C. Mason, M. Salmanian, H. Tang, HMS: Holistic MPR selection and network connectivity for tactical edge networks, in: Proceedings of the IEEE Military Communications Conference (MILCOM), 2015, pp. 726–731.

[20] J. Wu, F. Dai, Broadcasting in ad hoc networks based on self-pruning, in: Proceedings of the IEEE Conference on Computer Communications (INFOCOM), 3, 2003, pp. 2240–2250.

[21] F. Dai, J. Wu, An extended localized algorithm for connected dominating set formation in ad hoc wireless networks, IEEE Trans. Parallel Distrib. Syst. 15 (10) (2004) 908–920.

[22] L. Li, J.Y. Halpern, Minimum-energy mobile wireless networks revisited, in: Proceedings of the IEEE International Conference on Communications (ICC), 1, 2001, pp. 278–283.

[23] W. Heinzelman, A. Chandrakasan, H. Balakrishnan, Energy-efficient communication protocol for wireless microsensor networks, in: Proceedings of the Hawaii International Conference on System Sciences (HICSS), 8, 2000.

[24] Y. Xu, J. Heidemann, D. Estrin, Geography-informed energy conservation for ad hoc routing, in: Proceedings of the ACM International Conference on Mobile Computing and Networking (MobiCom), 2001, pp. 70–84.

[25] T. Opsahl, F. Agneessens, J. Skvoretz, Node centrality in weighted networks: generalizing degree and shortest paths, Soc. Netw. 32 (3) (2010) 245–251.

[26] Q. Tang, K. Yang, P. Li, J. Zhang, Y. Luo, B. Xiong, An energy efficient MCDS construction algorithm for wireless sensor networks, EURASIP J. Wirel. Commun. Netw. (1) (2012) 83.

[27] P.-J. Wan, K.M. Alzoubi, O. Frieder, Distributed construction of connected dominating set in wireless ad hoc networks, Mob. Netw. Appl. 9 (2004) 141–149.

[28] K.N. Patel, R.H. Jhaveri, A survey on emulation testbeds for mobile ad-hoc networks, Procedia Comput. Sci. 45 (2015) 581–591.

[29] F.A. Onat, I. Stojmenovic, Generating random graphs for wireless actuator networks, in: Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2007, pp. 1–12.

[30] J. Sum, J. Wu, K.I.-J. Ho, Analysis on a localized pruning method for connected dominating sets, J. Inf. Sci. Eng. 23 (2007) 1073–1086.

[31] J.Y. Yu, P.H.J. Chong, A survey of clustering schemes for mobile ad hoc networks, IEEE Commun. Surv. Tutor. 7 (1) (2005) 32–48.

[32] I. Rubin, F. Baiocchi, A. Cuomo, P. Salvo, Vehicular backbone network approach to vehicular military ad hoc networks, in: Proceedings of the IEEE Military Communications Conference (MILCOM), 2013.

[33] A.A. Aziz, Y.A. Sekercioglu, P. Fitzpatrick, M. Ivanovich, A survey on distributed topology control techniques for extending the lifetime of battery powered wireless sensor networks, IEEE Commun. Surv. Tutor. 15 (1) (2013) 121–144.

[34] J.P. Macker, I.J. Taylor, Prediction and planning of distributed task management using network centrality, in: Proceedings of IEEE MILCOM, 2014, pp. 859–864.
[35] J. Duan, D. Gao, C.-H. Foh, H. Zhang, TC-BAC: a trust and centrality degree based access control model in wireless sensor networks, Ad Hoc Netw. 11 (8) (2013) 2675–2692.
[36] L. Maccari, Q. Nguyen, R. Lo Cigno, On the computation of centrality metrics for network security in mesh networks, in: Proceedings of the IEEE Global Communications Conference (GLOBECOM), 2016.
[37] L. Maccari, R. Lo Cigno, Pop-routing: centrality-based tuning of control messages for faster route convergence, in: Proceedings of the IEEE International Conference on Computer Communications (INFOCOM), 2016.
[38] N. Magaia, A.P. Francisco, M. Pereira P. Correia, Betweenness centrality in delay tolerant networks: a survey, Ad Hoc Netw. 33 (2015) 284–305.
[39] S. Boccaletti, G. Bianconi, R. Criado, C.I. del Genio, J. Gomez Gardenes, M. Romance, I. Sendina-Nadal, Z. Wang, M. Zanin, The structure and dynamics of multilayer networks, Phys. Rep. 544 (2014) 1–122.
[40] M. Kitsak, L.K. Gallos, S. Havlin, F. Liljeros, L. Muchnik, H.E. Stanley, H.A. Makse, Identification of influential spreaders in complex networks, Nat. Phys. 6 (2010) 888–893.

Dimitrios Papakostas is a Ph.D. student at the Department of Electrical and Computer Engineering at the University of Thessaly, Greece. His research interests lie in the area of military ad hoc networks, and network science.

Soheil Eshghi is a postdoctoral researcher at the Department of Electrical Engineering at Yale University. Prior to that he was a postdoctoral student at Cornell University. His research interests are in optimal control, stochastic optimization, network epidemiology. Eshghi received a Ph.D. in electrical engineering from the University of Pennsylvania, Philadelphia, PA, USA.

Dimitrios Katsaros is a visiting assistant professor in the Department of Electrical Engineering at Yale University, and an assistant professor in the Department of Electrical and Computer Engineering at the University of Thessaly, Greece. His research interests include distributed systems. Katsaros received a Ph.D. in informatics from the Aristotle University of Thessaloniki, Greece.

Leandros Tassiulas is a professor in the Department of Electrical Engineering at Yale University. His research interests include computer and communication networks. Tassiulas received a Ph.D. in electrical engineering from the University of Maryland, College Park. He is an IEEE Fellow.