2019 American Control Conference (ACC)
Philadelphia, PA, USA, July 10-12, 2019

Energy-Aware Distributed Edge Domination of Multilayer Networks

Dimitrios Papakostas, Soheil Eshghi, Dimitrios Katsaros, Leandros Tassiulas

***Abstract*— Monitoring and maintaining communications in a multilayer ad hoc wireless network requires a stable communication overlay. Such monitoring should be resilient, avoiding reliance on a single (or a few) layers, and energy-aware, not being vulnerable to the exhaustion of available energy in a single network element. Thus, there is possibly a trade-off between overlay size, interlayer connectivity, and the energy distribution within the overlay. In this paper, we present three distributed energy-aware multilayer connected edge dominating set algorithms, show how they manage such trade-offs in practical scenarios, and show that CCEDS, a pruned centrality-based distributed algorithm, has the best performance.**

I. Introduction

Modern military battlefields consist of an increasing array of entities with wireless communication and sensing capabilities. In previous work [1], such heterogeneous allied systems have been modeled as multilayer ad hoc networks, in which each layer represents a type of battlefield entity (e.g., helicopters, UAVs, infantry); such multilayer tactical networks may arise in other settings as well, e.g., [2]. Designing networks with high numbers of inter-layer links immunizes the network to (possibly correlated) failures in any particular layer.

While the increased number of layers and inter-layer links increases the resilience of such networks, it complicates network functions such as routing and scheduling. To accomplish these tasks successfully in the long run, the network must be able to sense changes in topology, and specifically link failures, efficiently, with the sensing overlay itself being resilient to the same failures. This necessitates the design of resilient network overlays for either network management/monitoring or data forwarding, as the communication among different layers cannot be allowed to break easily (accidentally or due to malicious attacks). The sensed information gathered (and aggregated) by such a network overlay can be used in both centralized and decentralized control of the aforementioned network functions.

Distributed computation of a resilient network overlay for communication link monitoring in single layer ad hoc networks has been well studied, e.g., [3], [4]; This is different from traditional distributed approaches to ad hoc routing as routing algorithms are designed to minimize latency (thus seeking shortest-paths) or increase availability and prevent network choking (thus selecting multiple paths to the destination).

The computation of a resilient network overlay for multilayer ad hoc networks is significantly harder to address, especially using distributed algorithms, because coordination failures can lead to loss of communication or over-dependence on a specific layer. A first effort towards achieving the goal of building resilient overlays for multilayer ad hoc networks introduced custom-designed locally-computable centrality metrics [1]. However, neither the aforementioned work nor any others have considered the energy limitations of network entities in designing resilient overlays for link monitoring in multilayer ad hoc networks. Even though some network entities might be energy-rich, e.g., vehicles, others may face severe energy limitations, e.g., sensors, UAVs. Recharging batteries may be very difficult for entities deployed in a battlefield, as it both consumes time and detracts attention from the mission at hand. Therefore, the overlay should be built in such a way that all included entities have enough energy to keep the overlay operational and connected for as long as possible. Such energy-aware overlay creation is the goal of this paper.

However, the algorithms in [1] cannot be adopted here, because the current problem calls for solutions that incorporate as many *links* among different layers as possible into the overlay with the goal of increasing resiliency. Therefore, our problem cannot be described in terms of calculating a connected node dominating set for multilayer networks (as it was in [1]).

A. Motivating example

As an example, consider the two-layer network shown in Fig. 1, with the layers representing infantry and airborne units, and the node weights in parentheses denoting energy levels – the network is comprised of nodes S_i and C_i, and black links (those between nodes of the same layer) and red ones (those between nodes of different layers) links. It is assumed that communications between any two entities can traverse any path, and that monitoring the communications on an edge requires that we monitor at least one of its associated vertices.

We can monitor all links in such a network by picking an Edge Dominating Set (EDS).[1] However, a communication overlay requires communication and coordination, not just a monitoring capability, and thus we are interested in

D. Papakostas is with the Department of Electrical & Computer Engineering, University of Thessaly, Volos, Greece:`papdimit@inf.uth.gr`

D. Katsaros is with KIOS Research and Innovation Center of Excellence, University of Cyprus, Cyprus. Part of this work was done while he was a visiting assistant professor at the Department of Electrical Engineering at Yale University: `katsaros.dimitrios@ucy.ac.cy`

S. Eshghi and L. Tassiulas are with the Yale Institute of Network Science (YINS) and Department of Electrical Engineering, Yale University, New Haven, CT, USA: {`soheil.eshghi`, `leandros.tassiulas`}`@yale.edu`

The work of L. Tassiulas was supported, in part, by the ARO under grant W911NF1810378.

[1] An EDS is a set of edges such that every edge within the network shares an endpoint with one of the edges of the chosen edge-set.

978-1-5386-7926-5/$31.00 ©2019 AACC

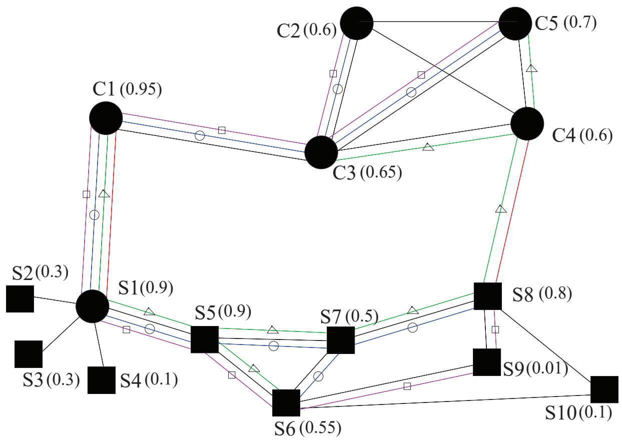

Fig. 1. A multilayer network with (square and circular) nodes connected by black (intralayer) and red (interlayer) links. Three different connected edge dominating sets, namely set of purple edges (with small squares on the links), set of blue edges (with small circles on the links) and set of green edges (with small triangles on the links) are also depicted.

creating a Connected EDS (CEDS). A CEDS allows the aggregation of monitored information, which is useful for situational awareness and sense-making by commanders in decentralized battlefields.

B. Objective

Our goal is to find a CEDS that is *small* in size, has many *interlayer links*, and which is *energy-aware*, in a *distributed* manner. Each of these goals, which are driven by practical concerns, impose limitations on the set of acceptable network overlays, leading to possible trade-offs:

1) *Cardinality*: A smaller CEDS limits the possible points of failure of the communication overlay.
2) *Number of interlayer links*: A resilient multilayer network overlay must limit dependence on any particular network layer while also limiting the likelihood of "islanding", i.e., the loss of communication among network layers. Increasing the number of interlayer links accomplishes these goals simultaneously.
3) *Energy awareness*: Communication and monitoring are both energy-intensive activities. The resulting energy depletion may exhaust the batteries of network elements, disconnecting the overlay and/or resulting in the loss of monitoring capabilities. In such settings, the CEDS must be created anew, a time- and energy-intensive process. Thus, the distribution of energy among elements chosen in the overlay must be such that there are fewer low-energy elements.
4) *Distributed computation*: The aggregation of information and the implementation of centralized decisions are major challenges in such ad hoc networks. A distributed algorithm, in which the network elements make determinations on their presence or absence in the overlay using local information, allows the creation and maintenance of the overlay in battlefields.

For example, in Fig. 1, each of the purple (small squares on links), blue (small circles on links) and green (small triangles on links) sets of edges comprise a (minimum cardinality) CEDS. The blue and purple overlays have the same number of inter-layer links, yet the edges in the blue overlay have end-points with higher energy levels, making it a preferable option. The green overlay is the best among the three, because it has more interlayer links and it is comprised of higher energy elements.

C. Contribution

We now clarify the difference between the present work and our own research that initiated the study of domination in multilayer networks, and also to highlight its main contributions.

In [1], [15] we analytically and experimentally investigated the problem of node domination for multilayer networks, whereas here we study edge domination for multilayer networks. In [6] – which is the companion paper of the present one – we introduced the problem of edge domination for multilayer networks, showed its relation to network control through the concept of maximal matching, gave complexity bounds, and developed heuristic algorithms. Here, we generalize the concepts of that paper by adding one more requirement, that of energy efficiency. This generalization presents challenges, because energy is not a feature of the edges, but of the network nodes, and thus new methods are needed to transform "node quantities" into "edge quantities".

In summary, the present article contributes a generalization of the problem introduced in [6], characterizes its complexity, provides simple and elegant methods to exploit node features, such as centrality and available energy, in order to quantify edge significance, and develops efficient heuristic algorithms to address the new problem. The rest of the paper is organized as follows: Section II formalizes the investigated problem; section III describes the proposed distributed heuristic algorithms to solve it; section IV analyzes their performance; and section V concludes the paper.

II. The EA-MCMCEDS problem

We first provide formal definitions of the domination concepts used to describe the overlay in single layer settings [5], and then factor in the effects of a multilayered architecture and energy-awareness, which set our problem, henceforth called the *Energy-Aware Minimum Connected Multi-Colored Edge Dominating Set* (EA-MCMCEDS), apart from the existing literature. The subsections proceed in the order of the four goals of the paper outlined in §I-B.

A. Minimum cardinality connected edge domination

For a single layer network $G = (V, E)$, an edge dominating set (EDS) is defined as follows:

Definition 1: An Edge Dominating Set (EDS) for G is a subset of the edge-set $E' \subseteq E$ such that for each edge $e \in E$, we either have $e \in E'$, i.e., e is a *dominating* edge, or there exists an edge $e' \in E'$ such that e' and e share a vertex, i.e., e is a *dominated* edge, *covered* by e'. A Minimum Edge Dominating Set (MEDS) of G is an EDS E' with the least possible cardinality $|E'|$.

As a consequence of this definition, for a vertex v, one of the following two possibilities holds: [Case (a)] v will be one of the two endpoints of an edge belonging to the EDS; in which case we will call it a *member of the overlay*, e.g., vertices $S6$ and $S7$ for either blue or green overlay in Fig. 1. [Case (b)] v, a *non-member of the overlay*, will be at one hop distance from a member of the overlay, e.g., vertex $S2$ for either blue or green overlay in Fig. 1. We can easily observe (by contradiction) that *all* vertices at a one-hop distance from a non-member of the overlay will be overlay members. This means that placing monitoring devices on the vertices (endpoints) of EDS edges will allow the monitoring of *all* communication within the system.

Overlay vertices can further be categorized into two groups: [Case a1] v will be a *core overlay member* if more than one EDS edge is incident to v; e.g., vertex $S7$ in Fig. 1 for either the blue or green overlay. [Case a2] Otherwise, if exactly one EDS edge is incident to v, it will be a *peripheral overlay member*; e.g., vertex $S6$ in Fig. 1 for either the blue or green overlay.

A further refinement to the EDS adds a notion of independence to the chosen edges:

Definition 2: An Independent Edge Dominating Set (IEDS) of G is an EDS E' such that no two edges within E' share an endpoint (vertex). An IEDS is also called a maximal matching [11].

Any minimum-cardinality IEDS (i.e., minimum maximal matching) will also be a Minimum Edge Dominating Set (MEDS) [12]. For an EDS of a certain cardinality (number of edges), the IEDS will require the most number of monitoring devices. However, there may exist some MEDSs requiring fewer monitoring devices than the minimum-cardinality IEDS [12].

While the EDS captures the goal of monitoring communications, it may not be able to provide the coordination and communication capability that is required of an overlay. For coordination within the chosen subset of edges, we introduce the notion of a Connected Edge Dominating Set (CEDS):

Definition 3: A Connected Edge Dominating Set (CEDS) of G is an EDS E' such that the edge-induced subgraph $G(E')$ is connected. The Minimum Connected Edge Dominating Set (MCEDS) is the CEDS with the minimum edge-cardinality.

B. Edge domination in multilayer networks

We now consider the equivalent of the single layer concepts described above in the context of multilayer networks, and describe how the relevant concepts described in §I-B can be captured in the creation of an overlay in such networks.

We first define (G^{ML}, E^{ML}) to represent a multilayer network, with $G^{ML} = \{G_1, G_2, \ldots, G_m\}$ such that:

- $G_i = (V_i, E_i)$ is a single layer network for all $1 \leq i \leq m$, with m being the number of layers.
- $E^{ML} = \{E_{ij} \subseteq V_i \times V_j | 1 \leq i, j \leq m, i \neq j\}$ is the set of existing interlayer links.

It follows that:

Definition 4: A CEDS of the multilayer network (G^{ML}, E^{ML}) is a set of edges $E'' \subseteq (\bigcup_{i=1}^{m} E_i) \bigcup E^{ML}$ such that the induced single layer subgraph on the vertex-set $\bigcup_{i=1}^{m} V_i$ is a CEDS.

The communication overlay should not be excessively reliant on any single layer, as it would be vulnerable to correlated failures. Thus, an ideal overlay would minimize the total number of edges within the overlay while at the same time maximizing the number of interlayer links.

Definition 5: A Minimum Connected Multi-Colored Edge Dominating Set (MCMCEDS) E'' of the multilayer network (G^{ML}, E^{ML}) is a CEDS with minimum cardinality $|E''|$ that has the maximum number of interlayer links, $|E'' \bigcap E^{ML}|$. Such a set is called multi-colored due to the practice of assigning a different non-black color to the elements of E^{ML} for each pair of layers connected by an edge, with all elements in $(\bigcup_{i=1}^{m} E_i)$ being considered black [6, Definition 7].

C. Energy availability and constraints

Key results within the field of ad hoc networks have pointed to the importance of factoring in the energy *distribution* among nodes, and not just the aggregate available energy, in network decisions. In particular, it has been shown that under certain circumstances, network lifetimes are maximized under equitable distributions of energy within the network, and that optimal communication policies put more of the communication burden on the network elements with the most remaining energy [9], [10]. Here, we assume that the energy available to each network element is known to that network element and can be communicated to its neighbors. Thus, we can adapt our overlay creation methods to be *energy-aware*:

Definition 6: An MCMCEDS creation method is Energy-Aware (EA) if it utilizes the energy available to network elements to find an overlay with high energy elements.
Ideally, all overlay nodes will have high energy, yet there is a possible trade-off between the cardinality and connectedness of the MCMCEDS and the energy of the overlay elements. To elaborate, given the results described from the ad hoc networks literature, our aim is to avoid elements with low energy in the overlay, even if they are well connected. Moreover, we seek to have higher energy nodes as *core* nodes, as such nodes have to relay information as well as performing the sensing and communication common to all overlay nodes. We further explore these concepts in our simulations (§IV-A).

D. Distributed energy-aware overlay generation

The nature of ad hoc networks demands that the overlay creation algorithm be distributed, with each element only having knowledge of their k-hop neighborhood. (Here, $k = 2$.) This adds the final element to the problem at the heart of this paper:

Problem 1 (Distributed EA-MCMCEDS Computation): We seek to find an energy-aware distributed algorithm that computes an MCMCEDS of a multilayer network given the energy available to network elements (i.e., vertices).

Proposition 1: Distributed EA-MCMCEDS computation is NP-hard.

It is easy to prove Proposition 1 following the reasoning of [6, Theorem 1], which involves a reduction from the NP-complete MCEDS problem [7, p. 102, Lemma 4.4.3]. Thus we will develop heuristic algorithms to solve the Distributed EA-MCMCEDS problem.

III. Proposed distributed algorithms

Here we describe three energy-aware distributed algorithms that heuristically solve the EA-MCMCEDS problem. The common principle in all the proposed algorithms is that when seeking which edges to include into the edge dominating set, these edges are selected based on their ability to a) dominate many other edges, b) connect different layers, and c) have energy-rich endpoints. Towards translating these goals into a heuristic rule for selecting edges, we use the local centrality measure we proposed in [1], *clPCI*, which we now enhance to take energy levels into account. Striving for simplicity and generalizability, along the lines of earlier works, e.g., [13], [15], we adopt a plain generalization of *clPCI*, termed *EclPCI*, for a node u with energy $E(u)$:

$$EclPCI(u) := E(u) \times clPCI(u). \tag{1}$$

The computation complexity of *EclPCI* index is $O(\Delta^2)$ in the worst case, where Δ is the maximum node degree in the network [15]. The value of *EclPCI* is used to prioritize nodes/edges. While other measures may quantify the same factors, *EclPCI* is the most natural one that combines energy availability of nodes with the computationally simple distributed centrality metric. Whenever edge-weights are needed, we define them to be the product of the *EclPCI* values of the endpoints (vertices) of the edge, so as to prioritize edges whose endpoints are *both* energy-rich vertices and possess strategic positions within the topology.

For the construction of the edge dominating set, the proposed algorithms either work with nodes to calculate node dominating sets and then turn them into edge dominating sets, or with edges (actually on their equivalent nodes in the *line graph* [5] of the original graph). In the interest of space, we give brief descriptions of the algorithms and provide their computation complexities and their pseudo-codes.

The first algorithm, CCEDS, calculates a connected node dominating set (CDS) and then applies a pruning mechanism using connectivity as quantified by *EclPCI* to establish a total order among nodes in the CDS. The set of edges whose endpoints are CDS nodes comprise the initial EDS. Finally, it adds into the EDS any edges that are attached to edges not incident to DS nodes. The details of the algorithm are shown in Algorithm 1.

Proposition 2: The computation complexity of CCEDS is $O(\Delta^3)$, where Δ is the maximum node degree in the network.

Proof: The worst-case construction phase of the CDS occurs 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. Further, the computation complexity of the respective pruning phase is also $O(\Delta^3)$ because a node u, in order to decide if it will act as a relay node or not, 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 $O(\Delta)$ complexity. Finally, the computation complexity of complementing the EDS with some "isolated" edges; i.e., that are not attached to a DS node, is $O(\Delta^2)$ because a relay node u needs to run once over its neighbor set of size $O(\Delta)$ and check for those of the neighbors that are not DS nodes whether they have a non-DS neighbor, and the neighbor set comparison again has $O(\Delta)$ complexity. ■

The second algorithm, EPEDS, first calculates an IEDS and then connects it. To elaborate, during the IEDS creation process, each node selects the highest-weight undominated edge incident to it and adds it to the EDS (if any exist). In order for the IEDS to be converted into a CEDS, each node that belongs to a DS edge adds enough incident edges (prioritized by the *EclPCI* of the one-hop neighbor at the other end of the edge) to collectively dominate its two-hop neighborhood. Finally, this algorithm uses a generic pruning policy which recognizes and then removes redundant edges with small weights from the CEDS. The details of the algorithm are shown in Algorithm 2.

Proposition 3: The computation complexity of EPEDS is $O(\Delta^3)$, where Δ is the maximum node degree in the network.

Proof: The worst-case construction phase of the EDS results when a host u has Δ neighbors and each one of them has Δ neighbors too. The adjacency matrix creation and its subsequent population with the weight value $w_{i,j}^{edge}$ of each respective edge requires a node u to compare its 1-hop and 2-hop neighbor set with $O(\Delta^2)$ neighbors in the worst case, and the neighbor set comparison has $O(\Delta)$ complexity. Further, the computation complexity of electing an edge as a DS edge is $O(\Delta^2)$, as a node u needs to compare its 1-hop neighbor set with Δ neighbors in the worst case, and the neighbor set comparison has a $O(\Delta)$ complexity. Connecting the EDS requires a host u 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. Finally, the computation complexity of the pruning phase is $O(\Delta^3)$, because a node u needs to calculate the coverage capability of a connected graph composed of both 1-hop and 2-hop neighbors in order to decide if it will act as a DS node or not. Thus, each node u compares its neighbor set with Δ^2 neighbors in the worst case, and the neighbor set comparison has a $O(\Delta)$ complexity. ■

The third algorithm, NPEDS, uses the mechanics of EPEDS to calculate the CEDS, and then uses the pruning mechanism of CCEDS.

The computational complexity of NPEDS is upper bounded by $O(\Delta^3)$ where Δ is the maximum degree in the network. This can be proved by mixing the relevant parts of the proofs of Propositions 2 and 3.

Algorithm 1: CCEDS

precondition : Known EclPCI index values of nodes in $(N(u)) \wedge (N^2(u))$
postcondition: Completed MCEDS election process
remarks : mlNetwork G =(V, E) where V and E are vertex & edge set, $R(u)$: relay node set of node u, $M(u)$ / $M(w_{i,j}^{edge})$: (T)rue/(F)alse indicator for node u / edge $w_{i,j}^{edge}$ being a DS node / edge.

repeat
  Add node $l \in N(u)$ with largest $EclPCI$ & which covers at least one new node in $N^2(u)$ to $R(u)$;
until *each node in* $N^2(u)$ *is covered by node(s) in* $R(u)$
Announce $R(u)$;
if *selected as a relay node* **then**
  $M(u) = T$;
  Announce status change;
  Build $S_{(u)}^{constrained} = u_1, u_2, \ldots, u_n \mid u_{k\ (1 \le k \le n)} \in N(u) \wedge N^2(u)$, $M(u_{k\ (1 \le k \le n)}) = T$, $EclPCI(u) < EclPCI(u_{k\ (1 \le k \le n)})$;
  if $S_{(u)}^{constrained}$ *is subject to* $N(u) \subset N(u_1) \cup N(u_2) ... \cup N(u_n)$ *and* $u_1, u_2, ..., u_n$ *form a connected graph* **then**
    $M(u) = F$; Set M($w_{i,j}^{edge}$) = F any edge $w_{i,j}^{edge}$ incident to node u; /* CDS Pruning */
    Announce status change;
    Return;
  end
  Build $S_{(u)}^{edge} = w_{u,l_1}^{edge}, w_{u,l_2}^{edge}, \ldots\ w_{u,l_m}^{edge} \mid w_{u,l_k\ (1 \le k \le m)}^{edge} \in$ E, $l_k \in$ N(u), $M(l_k)$ = F;
  if $\exists\ w_{u,l_k\ (1 \le k \le m)}^{edge} \in S_{(u)}^{edge}$ *adjacent to a non DS edge and that edge is not incident to a DS node* **then**
    Add $w_{u,l_k\ (1 \le k \le m)}^{edge}$ in the EDS ;
    Announce status change;
  end
end

Algorithm 2: EPEDS

precondition : Known EclPCI index values of nodes in $(N(u)) \wedge (N^2(u))$
postcondition: Completed MCEDS election process

Build edge adjacency matrix $E_{(u)}^{mat}$ with $N(u)$ & $N^2(u)$;
  /* $\exists\ e_{(i,j)} \in$ E $\iff i \in N(j) \wedge j \in N(i)$ */
Add weights $w_{i,j}^{edge} = EclPCI(i) * EclPCI(j)$ to $E_{(u)}^{mat}$;
Build $S_{(u)}^{edge} = w_{u,l_1}^{edge}, \ldots, w_{u,l_m}^{edge} \mid w_{u,l_k\ (1 \le k \le m)}^{edge} \in$ E, $l_k \in$ N(u);
if $\exists\ w_{u,l_k\ (1 \le k \le m)}^{edge} \in S_{(u)}^{edge}$ *not attached to DS edge* **then**
  Select the edge with the largest weight and set M($w_{u,l_k\ (1 \le k \le m)}^{edge}$) = T; /* EDS election */
  Announce status change;
end
repeat
  Select a node $l \in N(u)$ with the largest $EclPCI$ index value that covers at least one new node in $N^2(u)$;
  M(l) = T; M($w_{u,l}^{edge}$) = T; /* CEDS process */
until *each node in* $N^2(u)$ *is dominated by at least one DS node in* $N(u)$
Announce status of nodes in $N(u)$;
Build $S_{(u)}^{edge} = w_{u,l_1}^{edge}, \ldots\ w_{u,l_m}^{edge} \mid w_{u,l_k\ (1 \le k \le m)}^{edge} \in$ E, $l_k \in$ N(u), $M(l_k)$ = T;
repeat
  if $w_{u,l_k\ (1 \le k \le m)}^{edge}$ *is dominated by connected* $w^{edge} \in E_{(u)}^{mat}$ *with larger weight* **then**
    $M(w_{u,l_k\ (1 \le k \le m)}^{edge}) = F$; /* EDS Pruning */
    Announce status change;
  end
until *each* $w_{u,l_k\ (1 \le k \le m)}^{edge} \in S_{(u)}^{edge}$ *has been considered*

IV. Performance evaluation

Competing algorithms.
We compare the performance of the three proposed algorithms, CCEDS, EPEDS, and NPEDS, across the various aspects of the EA-MCMCEDS problem. Moreover, since degree-based node dominating set construction could be a viable technique, we developed WCEDS, which uses a straightforward energy-aware generalization of degree centrality [8] for multilayer networks, as a benchmark. WCEDS uses the same mechanics as CCEDS to calculate the CEDS, with the exception that the weighted degree centrality is used in place of *EclPCI*.

Datasets.
Due to the lack of publicly available, real-world tactical multilayer networks, we developed a generator for multilayer weighted networks which is described in detail in [14]. The construction of a multilayer network is controlled by the average degree of each node, by the number of nodes per layer (i.e., size of the layer), the diameter of each layer, and the number of layers. We apply four distinct *Zipfian* distributions, one per parameter of interest, controlled by the skewness parameter s of the respective Zipfian distribution:

- $s_{degree} \in (0, 1)$: to generate the frequency of highly interconnected nodes; therefore the degree distribution is Zipfian.
- $s_{layer} \in (0, 1)$: to choose how frequently a specific layer is selected; therefore the layer IDs collectively as edge anchors follow a Zipfian distribution.
- $s_{node} \in (0, 1)$: to choose how frequently a specific node is selected in a specific layer; therefore the distribution of node IDs as edge endpoints is Zipfian.
- $s_{weight} \in (0, 1)$: to choose how energy is distributed to nodes; therefore the energy distribution is Zipfian.

We selected a default setting for each of the parameters, calling them collectively the *skewness*. We represent them as a sequence of four floats, e.g., $0.5 - 0.5 - 0.5 - 0.5$, which means that $s_{degree} = 0.5$, $s_{layer} = 0.5$, $s_{node} = 0.5$ and $s_{weight} = 0.5$ (the default settings we used to create the datasets). Table I records all the independent parameters of our topology generator.

parameter	range	default
avg. node degree (D)	3, 6, 10, 15, 20	6
network diameter (H)	3, 5, 8, 12, 17	8
#network layers (L)	2, 3, 4, 5, 7	4

TABLE I

EXPERIMENTATION PARAMETERS VALUES.

A. Results

We performed a simulation-based performance evaluation of the competing algorithms in MATLAB. We include IEDS in the plots as a benchmark, but do not comment on its performance because it does not create a CEDS.

In Figs. 2-4, we plot the performance of the competitors in terms of the goals of the EA-MCMCEDS problem as the mean degree, diameter, and number of layers of the synthetic multi-layer network is varied. In these figures, the first row of histograms show the size of the CEDS that each algorithm creates, which is measured as a percentage of the total edges in the network, while the second row plots show the percentage of all the inter-layer links that are included in the CEDS. An ideal algorithm will minimize the former, while maximizing the latter. The third row of plots show the energy distribution of the vertices selected for the overlay, plotting their mean energy with associated error bars, while the fourth row of plots show the number of overlay nodes. An ideal algorithm would cover the network using relatively few, high-energy nodes, so it would maximize the third row of plots while minimizing the fourth.

1) Impact of topology density: In Fig. 2, we consider the impact of topology density on each competitor's performance. In the top row, we evaluate the size of the EDS that each competitor creates. We first observe that the size of the EDS is almost a decreasing function of node density, as in higher network densities, each node will participate in more edges, and each edge can thus dominate more edges. It is interesting that both CCEDS and WCEDS manage to create the smallest MCEDS (approximately 2.0 up to 2.1 times the size of IEDS) regardless of how sparse or dense the network is. On the other hand, EPEDS and NPEDS create the largest EDSs for mean degrees 3 and 6 (with $> 60\%$ and $> 50\%$ more edges, respectively), while they perform close to the other heuristics for larger mean degrees. The best performing algorithms in sparse and medium density networks start with an MCDS in creating an MCEDS, which means that they can start from much sparser overlay sets compared to methods that require the creation of a CEDS from the start, as without coordination, many edges are needed for domination in sparse networks.

In the second row, we see that CCEDS and NPEDS include more inter-layer links than competitors, with performance levels that do not change much with network density. On the other hand, while EPEDS is the best performing algorithm when mean degree = 3, its performance drops drastically for larger mean degrees. WCEDS also has a similar, yet less drastic, performance drop. This is because *EclPCI*, as used by CCEDS and NPEDS enables these algorithms to privilege inter-layer links for inclusion in

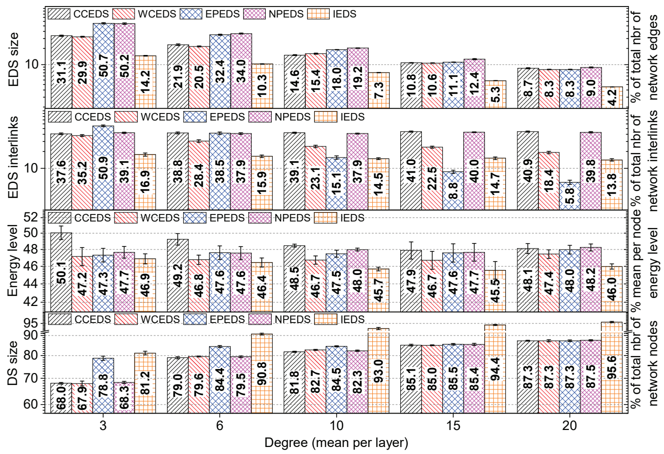

Fig. 2. Impact of network density on the performance of each competitor.

the CEDS, while WCEDS is layer-agnostic. The surprising drop in performance for EPEDS is due to its edge-pruning mechanism (as opposed to the node-pruning mechanism of CCEDS/NPEDS), which increases the likelihood of pruning inter-layer links: as the probability of the existence of a dominating edge in either of the linked layers increases with the mean degree, and the probability of self-deselection by the inter-layer edge in the pruning phase also increases.

In the third row, we see that CCEDS, on average, leads to a $3-5\%$ increase in overlay node energies in sparse networks (mean degrees 3 and 6) compared to the other competitors. This difference levels out at higher densities, except with the worst-performing WCEDS. This is tightly coupled with the arguments around the size of the selected EDS presented for the first row - CCEDS chooses fewer, yet more central and higher energy edges in sparse networks precisely because it does not impose creating a CEDS from the first step.

In the last row, we plot the relative size of the overlays. Interestingly, and counter-intuitively, we see that the overlay size grows with the mean degree. We may expect that a denser network could be dominated by fewer overlay nodes. However, the number of edges in a network grows with the edge density, and as we are seeking to build an *edge*-dominating set, the number of overlay nodes also grows. In this aspect as well, CCEDS, NPEDS, and WCEDS perform best, with EPEDS's edge-based pruning leading to a $> 14\%$ and $> 6\%$ handicap for mean degrees 3 and 6.

2) Impact of network diameter: In Fig. 3, we consider the impact of network diameter on each competitor's performance. In the top row, we observe that the size of the constructed EDS increases with the network diameter for all algorithms. This is the result of sparser neighborhoods, i.e., fewer links between network nodes. In other words, there are fewer, longer (in hops), and less distinct paths in the multilayer network, which leads to the election of a large number of edges for the EDS. Except for the worst-performing EPEDS (whose size goes from 29% larger than the best competitor when diameter $= 3$ up to 82% larger when diameter $= 17$) all the algorithms lead to similar EDS sizes. The pruning mechanism of EPEDS is responsible for its bad performance.

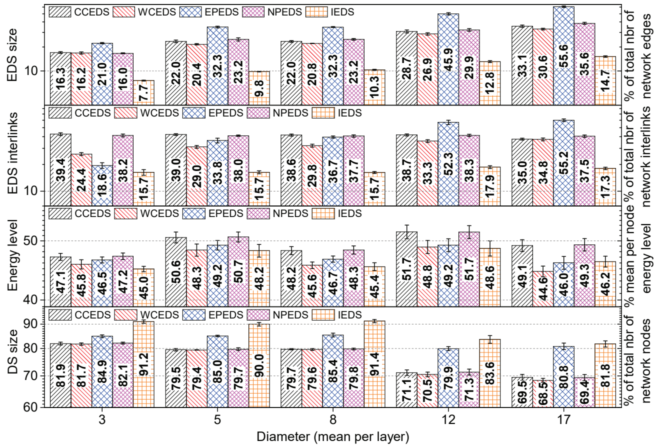

Fig. 3. Impact of network diameter on the performance of each competitor.

From the second row of plots, we see that CCEDS and NPEDS perform best overall with regards to the number of the EDS interlinks, with the relative number of EDS interlinks staying stable as the diameter is varied. This means that they both work well in bushy (small number of hops) or skinny (large number of hops) networks. EPEDS, on the other hand, performs remarkably well for larger network diameters, while performing poorly when diameter $= 3$ (where it includes less than the half number of interlinks in the EDS compared to CCEDS). For example, EPEDS creates an overlay with $> 45\%$ more interlinks compared to CCEDS and NPEDS when diameter $= 12$ or 17, yet this comes at the cost of a much larger EDS size, which is unacceptable. WCEDS, the layer-agnostic baseline, only reaches the respective performance of the other algorithms in terms of interlinks when diameter $= 17$, while having much fewer interlinks for more bushy networks. So, while WCEDS performs well in terms of EDS size for bushy networks, this is traded off against the lower resiliency of the resulting overlay. Note that the number of the interlinks that are included in the EDS by each algorithm is a direct consequence of the pruning mechanism they employ, and more precisely, how well each of them can distinguish between a simple edge and an interlink.

From the third row of plots, we see that all the algorithms lead to overlays with similar average energy levels and there is no clear winner. However, for medium-to-large network diameters (when it equals 8, 12 and 17) both CCEDS and NPEDS select nodes with slightly more energy (on average) into the overlay (showing a 3.5–10% improvement).

Interestingly, in the last row of plots, we see that the DS size decreases significantly for all the competitors when diameter $= 12$ and 17. This arises from the fact that the multilayer network is created from the interconnection of sparse and skinny networks in these settings. In such a case, the DS nodes are shared between many EDS edges. Most of the algorithms lead to similar DS sizes, except for EPEDS which leads to 3.5% (when diameter$= 3$) to 18.0% more nodes in the DS. This is due to its inefficient pruning mechanism.

3) Impact of the number of layers: In Fig. 4, we consider the impact of the number of network layers on each competitor's performance. From the top row of plots, we see that for the majority of the algorithms (all except EPEDS), the number of edges in the EDS decreases with an increase in the number of the multilayer network layers, leading to EDS selections approximately 2.0–2.4 times the size of the IEDS, because of the richer connectivity among layers' hub nodes imposed for the specific value(s) of Zipfian distribution(s). We also observe that all competitors lead to similar size EDS selections, except for EPEDS whose EDS is 30% larger than the rest.

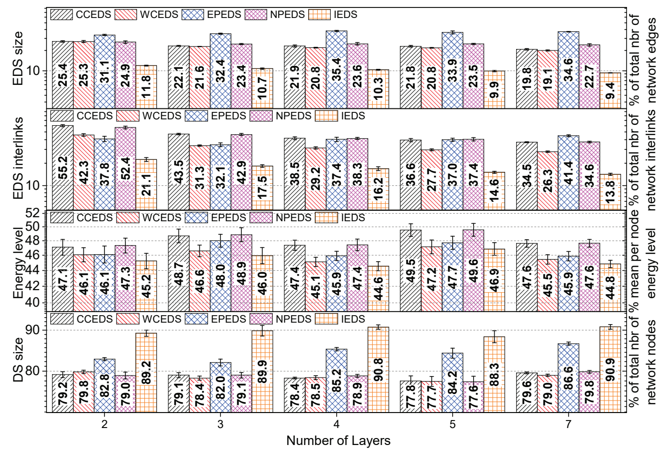

Fig. 4. Impact of number of layers on the performance of each competitor.

.

From the second row of plots, we observe that the number of EDS interlinks decreases as the number of layers increases. This is because it is increasingly difficult for all the algorithms to distinguish between interlayer and intralayer edges when the total number of edges increases (a by-product of the increase in the number of layers). However, both CCEDS and NPEDS manage to confront this problem more efficiently than the competition when we have 2-layer and 3-layer networks, with both having at least 25% and 38% more interlinks in the EDS than the competition, respectively. This is because EclPCI can improve the ability of an algorithm to distinguish between interlinks and intra-layer edges. It is interesting to note that the relative differences in the number of interlinks chosen by the competitors does not change with the number of network layers, except in the case of the 7-layer network, in which EPEDS chooses a relatively large number of interlinks in its relatively large EDS. Again, WCEDS has the worst performance among the competitors in this aspect due to its layer-agnostic nature, except for the 2-layer network where it performs 12% better than EPEDS.

In the third row of plots, we see that both CCEDS and NPEDS create overlays with the most average energy per node irrespective of the number of network layers. Interestingly, the number of network layers has no effect on the relative differences in the performance of the competitors.

Finally, in the bottom row of plots, we observe that the percentage of nodes included in the DS does not change with the number of network layers for the majority of the competitors. Once again, the exception is EPEDS, for which the number of DS nodes increases with the number of layers.

4) Energy analysis of the overlay: In Fig. 5, we analyze the (average) energy levels of the core, periphery, and non-members of the overlay along with the size of the EDS for the competing algorithms, in order to illustrate the tradeoff between picking only high-energy nodes and providing sufficient coverage. Each bar has three colored segments: a pink segment corresponding to core overlay nodes, a light orange segment corresponding to peripheral overlay nodes, and a light green segment corresponding to non-members of the overlay. The height of each colored segment represents the percentage of nodes belonging to that node class; therefore, the sum of the heights of the segments is 100%. The number superimposed within each colored segment depicts the average energy of the nodes belonging to that class.

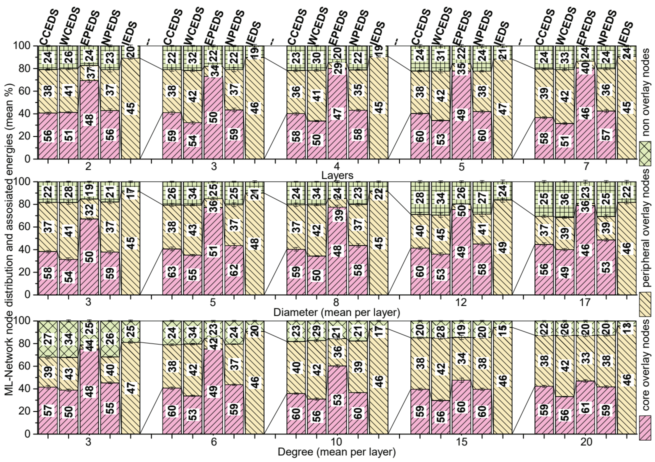

Fig. 5. Energy levels (average) of the core, periphery, and non-members of the overlay along with the size of the EDS.

An algorithm will be efficient in terms of overlay size if the total height of the associated pink and the light orange segment is small relative to its competitor(s). We observe from the plots (which vary the number of layers, the diameter, and the mean degree) that CCEDS produces the smallest overlay in (almost) all cases, as observed in earlier figures as well.

An algorithm is efficient in terms of energy, *primarily* if the average energy of its core overlay nodes (superimposed number of the pink segment) is high and the average energy of the overlay's non-members (super-imposed number over the light green segment) is low. We observe that on average, CCEDS's core overlay nodes have higher energy than those of their competing algorithms and outperform the second-best algorithm, NPEDS, by around 4% in some cases. However, NPEDS has quite similar performance to CCEDS in general, and is slightly superior to CCEDS for networks with very small diameter (1.5% improvement).

V. Conclusions

We considered distributed methods of creating a minimum-size overlay network of monitoring devices over a wireless multilayer ad hoc network. We emphasized the overlay network's resilience to correlated layer failures and to energy depletion in devices, paying special attention to inter-layer links, and formalized the problem in terms of edge dominating sets (EDS) in multilayer networks (EA-MCMCEDS).

We proposed three distributed algorithms for solving EA-MCMCEDS, namely CCEDS, EPEDS, and NPEDS. CCEDS creates a connected EDS by first starting from a node dominating set, whereas the other two start from an independent EDS. All employ smart pruning heuristics to reduce the size of the resulting connected EDS. We compared their performance, along with that of a baseline competitor, using extensive simulations varying network topological characteristics and energy distribution patterns in synthetic multilayer networks. CCEDS was, by far, the best performing algorithm in terms of EDS cardinality, layer-based resilience, and energy composition in (almost) all cases.

While our heuristics performed well over a range of scenarios, investigating the approximability of the EA-MCMCEDS problem is an important line of future work.

References

[1] D. Papakostas, P. Basaras, D. Katsaros, and L. Tassiulas, "Backbone formation in military multilayer ad hoc networks using complex network concepts," in *Proceedings of IEEE MILCOM*, 2016, pp. 842–848.

[2] Y.E. Sagduyu, Y. Shi, T. Erpek, S. Soltani, S.J. Mackey, D.H. Cansever, M.P. Patel, B.F. Panettieri, B.K. Szymanski, G. Cao, "Multilayer MANET routing with social-cognitive learning," in *Proceedings of IEEE MILCOM*, 2017, pp. 103–108.

[3] E. Fata, S.L. Smith, S. Sundaram, S., "Distributed dominating sets on grids," in *Proceedings of American Control Conference (ACC)*, 2013, pp. 211–216.

[4] H. Kazemi, G. Hadjichristofi, L.A. DaSilva, "MMAN - a monitor for mobile ad hoc networks: design, implementation, and experimental evaluation," in *Proceedings of ACM WinTECH*, 2008, pp. 57–64.

[5] T. W. Haynes, S. Hedetniemi, and P. Slater, *Fundamentals of Domination in Graphs*, ser. Chapman & Hall/CRC Pure and Applied Mathematics. CRC Press, 1998.

[6] D. Papakostas, S. Eshghi, D. Katsaros, and L. Tassiulas, "Distributed algorithms for multi-layer edge-dominating sets," *IEEE Control Systems Letters*, vol. 3, no. 1, pp. 31-36, 2019.

[7] A. Munaro, "On some classical and new hypergraph invariants," *PhD Thesis*, Université Grenoble Alpes, 2016.

[8] T. Opsahl, F. Agneessens,and J. Skvoretz, "Node centrality in weighted networks: Generalizing degree and shortest paths," *Journal of Social Networks*, vol. 32, no. 3, pp. 245–251, 2010.

[9] J.-H. Chang, L. Tassiulas, "Energy conserving routing in wireless ad-hoc networks," in *Proceedings of IEEE INFOCOM*, 2000, pp. 22–31.

[10] S. Eshghi, M. Khouzani, S. Sarkar, N. Shroff, and S. Venkatesh, "Optimal energy-aware epidemic routing in DTNs," *IEEE Transactions on Automatic Control*, vol. 60, no. 6, pp. 1554–1569, 2014.

[11] T. Fujito, "On approximability of the independent/connected edge dominating set problems," *Information Processing Letters*, vol. 79, no. 6, pp. 261–266, 2001.

[12] M. Yannakakis and F. Gavril, "Edge dominating sets in graphs," *SIAM Journal of Applied Mathematics*, vol. 38, no. 3, pp. 364–372, 1980.

[13] N. Dimokas, D. Katsaros, "Detecting energy-efficient central nodes for cooperative caching in wireless sensor networks," *Proceedings of IEEE AINA*, 2013, pp. 484-491.

[14] P. Basaras, G. Iosifidis, D. Katsaros, and L. Tassiulas, "Identifying influential spreaders in complex multilayer networks: A centrality perspective," *IEEE Transactions on Network Science and Engineering*, vol. 6, no. 1, pp. 31–45, 2019.

[15] D. Papakostas, S. Eshghi, D. Katsaros, L. Tassiulas, "Energy-aware backbone formation in military multilayer ad hoc networks," *Ad Hoc Networks*,vol. 81, pp. 17–44, 2018.