IEEE CONTROL SYSTEMS LETTERS, VOL. 3, NO. 1, JANUARY 2019

Distributed Algorithms for Multi-Layer Connected Edge Dominating Sets

Dimitrios Papakostas, Soheil Eshghi, Dimitrios Katsaros, and Leandros Tassiulas

***Abstract*—Monitoring the state of communications in a distributed multilayer network with differing node capabilities requires the maintenance of a backbone which is a connected edge dominating set. In this letter, we present distributed algorithms that can efficiently create such multilayer resilient connected edge-dominating sets. After establishing the complexity of the problem and our proposed heuristics, we experimentally compare their performance while varying multiple characteristics of the underlying networks.**

***Index Terms*—Edge dominating sets, monitoring, backbone, ad hoc networks, multilayer networks.**

I. Introduction

The distributed nature of modern networks and the limited processing power of networked sensors and embedded systems used in Internet of Things (IoT) applications has led to new security vulnerabilities [1]. Intruders can inject malicious communications between any two networked elements without aiming to have the message propagated to any further target. The increasing variety, capability, and complexity of network elements has increased this risk. Furthermore, the evolution of these networks leaves them vulnerable to errors and compatibility issues when new elements are added to the network. While these issues may be sensed by the communicating network elements, their limitations do not allow them to compute remedies, necessitating communication to elements with more processing power. In this letter, we present a framework for monitoring network failures using connected edge dominating sets in multilayer networks, and then we provide efficient distributed algorithms for their computation.

Manuscript received March 6, 2018; revised May 30, 2018; accepted June 10, 2018. Date of publication June 27, 2018; date of current version July 13, 2018. The work of S. Eshghi and L. Tassiulas was supported by U.S. Office of Naval Research under Award N00014-14-1-2190. Recommended by Senior Editor J. Daafouz. *(Corresponding author: Soheil Eshghi.)*

D. Papakostas is with the Department of Electrical and Computer Engineering, University of Thessaly, 38221 Volos, Greece (e-mail: apdimit@inf.uth.gr).

S. Eshghi and L. Tassiulas are with the Department of Electrical Engineering, Yale Institute of Network Science, Yale University, New Haven, CT 06520-8263 USA (e-mail: soheil.eshghi@yale.edu; leandros.tassiulas@yale.edu).

D. Katsaros is with the Department of Electrical and Computer Engineering, University of Thessaly, 38221 Volos, Greece, and also with the Department of Electrical Engineering, Yale University, New Haven, CT 06520-8263 USA (e-mail: d.katsaros@yale.edu).

Digital Object Identifier 10.1109/LCSYS.2018.2850921

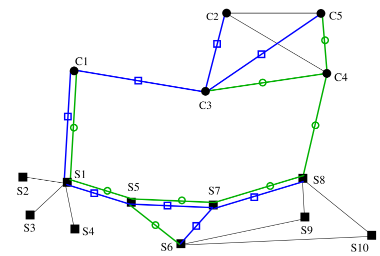

Fig. 1. Two minimum connected edge dominating sets: the blue (with square marks) includes one inter-layer edge, and the green (with circular marks) includes two inter-layer edges.

Consider the case where we wish to be able to monitor all the communication taking place among nodes of a wireless ad hoc network such as the one shown in Fig. 1. It is assumed that any pair of nodes can initiate an exchange of packets and the routing may follow any path of the network, e.g., not only the shortest-path route between the communicating nodes. In principle, this task requires us to recognize a set of edges (communication links) such that every other edge is adjacent to at least one edge belonging to this set; then, by placing monitoring devices at the endpoints of each edge belonging to this set we can achieve our goal. Such a set of edges is termed an *edge dominating set (EDS)* in graph-theoretic terms. Due to cost considerations, we are interested in identifying such sets with minimum cardinality, i.e., we seek *minimum edge dominating sets (MEDS)*. However, as it is often the case for ad hoc networks, the set of monitoring devices must be able to output any intercepted information; therefore the MEDS must be *connected* (MCEDS), and, moreover, must be computed in a distributed fashion. Looking at Fig. 1, we can confirm that the set of blue edges constitutes a MCEDS, and also the set of green edges constitutes a MCEDS.

The concept of *network layers* can capture the diversity in the capabilities of network elements, as well as their differing roles. For example, although traditional ad hoc networks are treated as single layer networks, military tactical ad hoc networks [2] are considered to be *multilayer* networks due to the existence of different types of units (infantry, vehicles or airborne units), where nodes belong to different layers, i.e., groups. For instance, in Fig. 1 the node set $C1$–$C5$ comprise one layer and nodes $S1$–$S10$ comprise another layer.

Finding an MCEDS for multilayer networks is somewhat more complicated than calculating MCEDS for single layer networks, both for technical reasons (see Theorem 2), and for application-specific reasons, e.g., robustness. Looking again at the blue and green MCEDS's in Fig. 1, we observe that the green one includes two edges that connect the different layers (inter-layer edges), whereas the blue only has one such edge. Increasing the number of inter-layer edges can improve the network's resiliency to failures in any particular layer [2].

In this letter, we cast our monitoring problem for multilayer networks, which entailed finding an MCEDS in a distributed manner containing many inter-layer edges, into a new form of generic domination problems. We name this problem the *multi-colored minimum connected edge dominating set problem* (MCMCEDS), and we will describe it here in terms of calculating the minimum multi-colored edge dominating set. The framework and the algorithms proposed can be used for efficiently detecting and avoiding interference conditions in large wireless IoT networks, or even in more specialized setting such as those enabling dynamic frequency selection (DFS) where radar signals must be detected and protected against interference from 5GHz radios; dominating sets concepts have been used in the past for monitoring problems [3], [4].

The contributions of this letter are as follows:

- It introduces the novel problem of finding a (minimum) connected edge dominating set in multilayer networks with the additional goal of including many inter-layer links into the EDS (Section II). This problem extends ideas related to those developed in [5].
- It analyzes its computational complexity (Section III).
- It proposes three heuristic distributed algorithms for it (Section IV).
- It proves an analytic result that relates the cardinality of an independent edge dominating to the cardinality of a corresponding connected edge dominating set (Section IV-B).
- It conducts a performance evaluation of the proposed algorithms against two baseline competitors (Section V).

We define the MCMCEDS problem in Section II. We then present results on the complexity of MCMCEDS computation in Section III, and discuss our approaches to computing heuristics and their rationale Section IV. We present extensive simulation results in Section V. We survey related work on Section VI.

II. The MCMCEDS Problem

A. Edge Domination in Traditional Settings

Firstly, we will provide some basic definitions on dominating sets [3] before we formulate this letter's problem.

Definition 1: An edge dominating set $EDS(G)$ of a network (G, E) (G is the set of nodes, and E is the set of edges) is any subset of E such that any edge $e \in E$ is either a member of $EDS(G)$ (it is a *dominating* edge) or it has one common endpoint with at least one dominating edge (it is a *dominated* edge).

Let x_e be an indicator variable representing whether $e \in E$ is included in $EDS(G)$. Therefore, Definition 1 is equivalent to saying that for each $e \in E : x_e + \sum_{e' \in N(e)} x_{e'} \geq 1$, where $N(e)$ is the set of neighboring edges of edge e (i.e., those with one common endpoint). Note that in the *line-graph* $L(G)$ of graph G, in which every edge is replaced with a vertex and vice versa, and the incidence relationship between edges and vertices is preserved [6], an edge dominating set in G, $EDS(G)$, is translated to a *dominating set* (DS).

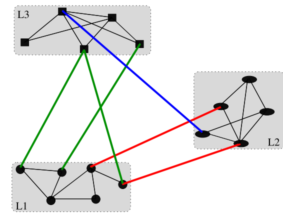

Fig. 2. A multicolored multi-layer network with 3 layers (L1, L2, L3).

Definition 2: An independent edge dominating set $IEDS(G)$ of a network (G, E) (also referred to as a maximal matching [7]) is any edge dominating set of G such that no two edge dominators share an endpoint.

Definition 3: A connected edge dominating set $CEDS(G)$ of a network (G, E) is any edge dominating set of G such that the set of dominating edges along with their endpoints comprise a connected network.

The line-graph $(L(G))$ preserves connectivity [6], so in translation, $CEDS(G)$ becomes a Connected Dominating Set (CDS) of the line-graph.

Definition 4: A minimum connected edge dominating set $MCEDS(G)$ of a network (G, E) is any $CEDS$ of G with the additional property that it contains the least possible number of dominating edges.

At this stage, the link between the $CEDS$ and its equivalent in the line-graph is broken: an $MCEDS(G)$ will translate to a CDS with the minimum number of nodes, and not edges, in the line-graph. Therefore, the problems of finding the Minimum Connected Dominating Set ($MCDS$) [8] and the $MCEDS$ are not linked in a straightforward manner. So an $MCDS$ in $L(G)$ will be a $CEDS$ in G, but there is no guarantee that its cardinality will be minimal.

B. Edge Domination in Multi-Layered Network Settings

Definition 5: A multi-layer network comprised 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), as defined earlier, and $E^{ML} = \{E_{i,j} \subseteq G_i \times G_j; i, j \in \{1, \ldots, n\}, i \neq j\}$ is a set of inter-layer edges.

Definition 6: A minimum connected edge dominating set of a multi-layered network $MCEDS(G^{ML})$ includes the minimum set of edges such that their *induced* subgraph is connected and edges not in this set are adjacent to at least one edge within it.

In Fig. 1, $G_1 = \{Si, i = 1, \ldots, 10\}$, $G_2 = \{Ci, i = 1, \ldots, 5\}$, and E^{ML} is the set of all edges connecting them, e.g., $\langle S1, C1 \rangle$.

Definition 7: An edge-multicolored multi-layer network (see Fig. 2) is a multi-layer network with these two properties:

p-1) all edges e whose endpoints both belong to a single (any) layer, i.e., $e \in E_i, \forall i \in \{1, \ldots, n\}$ have the same color (black), e.g., black edges in Fig. 2.

p-2) all edges l whose endpoints belong to different layers, i.e., $l \in E_{i,j} \subseteq G_i \times G_j, i, j \in \{1, \ldots, n\}, i \neq j$ will have the same color, which is different from the color of edges $c \in E_{x,y} \subseteq G_x \times G_y, x, y \in \{1, \ldots, n\}, [x, y] \neq [i, j]$, e.g., red edges in Fig. 2.

Definition 8: A multi-colored minimum connected edge dominating set of a multilayer network $MCMCEDS(G^{ML})$ is an $MCEDS(G^{ML})$ with the maximum number of colorful (i.e., non-black) edges.

Problem 1 (dist-MCMCEDS): We seek to find an *MCMCEDS*(G^{ML}) for a multi-layer network G^{ML} in a distributed fashion, i.e., having only knowledge of the k-hop neighborhood around each node. Here, we set $k = 2$.

III. Complexity of the MCMCEDS Problem

Theorem 1: The MCMCEDS problem is NP-hard.

Proof: Assume we have a single-layer graph $G = (V, E)$ and we seek to find its *MEDS*. Now, create a 2-layer network (G^{ML}, E^{ML}), where $G^{ML} = \{G^i, i = 1, 2\}$ have the same vertices as V, and a set of inter-layer edges $E^{ML} = \{E_{i,j} \subseteq G_i \times G_j; i, j \in \{1, \ldots, n\}, i \neq j\}$ by assigning one edge in E to E_{ML} and the rest to E_1, and E_2 uniformly at random. If *MCMCEDS* for such a (G^{ML}, E^{ML}) was not NP-hard, we could use it at most $|E|$ times (varying the edge assigned to E_{ML}) to find a solution to *MCEDS*, a known NP-complete problem [9, p. 102, Lemma 4.4.3]. ■

IV. Heuristics for the MCMCEDS Problem

Since our problem is NP-hard, we wish to design heuristic algorithms that can encapsulate the idea of including as many inter-layer edges as possible into the EDS. In our previous work [2], [10] we have introduced the family of the Power Community Index *(PCI)* centrality measures for multilayer networks, namely *mlPCI* and *clPCI*, whose purpose is to assign a value to each node which depicts its connectivity both to its layer and to other layers. In [2] we used *clPCI* and *mlPCI* for the purpose of establishing a backbone for multilayer ad hoc networks based on the calculation of a node dominating set. Note that a simple application of these algorithms to create a connected node dominating set is insufficient, as it may leave some edges undominated. We will not repeat the definitions, but instead give the distributed algorithms for the calculation of the edge dominating sets, and calculate their computational complexities as a function of Δ, the maximum node degree in the network.

A. PCI Approaches

In Algorithm 1, lines (1)–(9) are distributed and executed by every node u in order to select which of the edges incident on it (i.e., on u) will be included in the IEDS. The selection is based on such a multilayer centrality measure. Since the centrality measure has been defined for nodes and not for edges, we use the product 'value' of each edge's end-nodes to define the edge's value. The fictitious operation of line (10) unites every node's selection in order to construct the final IEDS. The proof of algorithm's correctness, in the sense that it constructs an IEDS is very similar to that reported in [11, Th. 4.2] and thus it will be omitted for all algorithms presented.

Algorithm 1: IEDS

postcondition: Completed IEDS election process
remarks : multilayer network G=(V,E), $S^{edge}_{(u)}$: edges incident to u, $M(u)$ / $M(w^{edge}_{i,j})$: True(T) / False(F) indicator for node u / edge $w^{edge}_{i,j}$ being a DS node / edge.

1 Identification of 1-hop ($N(u)$) and 2-hop ($N^2(u)$) neighborhood via distributed beaconing and calculation of clPCI indexes of the nodes;
2 Build local edge adjacency matrix $E^{mat}_{(u)}$ with $N(u)$ & $N^2(u)$;
/* $\exists\ e_{(i,j)} \in$ E $\iff$ $i \in N(j) \wedge j \in N(i)$ */
3 Add weights $w^{edge}_{i,j} = clPCI(i) * clPCI(j)$ to $E^{mat}_{(u)}$;
4 Build $S^{edge}_{(u)} = w^{edge}_{u,l_1}, \ldots, w^{edge}_{u,l_m} \mid w^{edge}_{u,l_k} \in$ E, $l_k \in$ N(u) $\forall_{k \leq m}$;
5 **if** $\exists\ w^{edge}_{u,l_k\ (1 \leq k \leq m)} \in S^{edge}_{(u)}$ *not attached to DS edge* **then**
6   Select the edge with the largest weight and set M(u) = T;
7   M($w^{edge}_{u,l_k\ (1 \leq k \leq m)}$) = T; /* EDS election */
8   Announce status change;
9 **end**
10 Collect all edges (across the network) with a status=T;

Algorithm 2: MLEDS#1

precondition : Completed IEDS election process
postcondition: Completed MCEDS election process
remarks : $R(u)$: relay node set of node u.

1 **If** $M(u) = F$ **then** Return; /* not a DS node */
2 **repeat**
3   Add in $R(u)$ a node $l \in N(u)$ with the largest *clPCI* index that covers at least one new node in $N^2(u)$;
4   M(l) = T; M($w^{edge}_{u,l}$) = T; /* CEDS process */
5 **until** *each node in* $N^2(u)$ *is covered by node(s) in* $R(u)$
6 Announce status change;
7 Build $S^{edge}_{(u)} = w^{edge}_{u,l_1}, \ldots\ w^{edge}_{u,l_m} \mid w^{edge}_{u,l_k\ (1 \leq k \leq m)} \in$ E, $l_k \in$ N(u), $M(l_k)$ = T;
8 Sort $S^{edge}_{(u)}$ in increasing order of the w^{edge} weights.
9 **repeat**
10   **if** $w^{edge}_{u,l_k\ (1 \leq k \leq m)}$ *is dominated by connected* $w^{edges} \in E^{mat}_{(u)}$ *with larger weight* **then**
11         $M(w^{edge}_{u,l_k\ (1 \leq k \leq m)})$=$F$; /* EDS Pruning */
12   Announce status change;
13   **end**
14 **until** *each* $w^{edge}_{u,l_k\ (1 \leq k \leq m)} \in S^{edge}_{(u)}$ *has been considered*
15 Collect all edges (across the network) with a status=T;

Proposition 1: The computation complexity of IEDS is $O(\Delta^2)$ in the worst case.

Proof: The worst case computation complexity of IEDS selection is when a node u has Δ neighbors and each one of them has Δ neighbors too. During the build-up of the edge adjacency matrix, node u needs to compare its 1-hop and 2-hop neighbor set with Δ^2 neighbors in the worst case, and the neighbor set comparison has a $O(\Delta)$ complexity. The same computation cost applies to the population of the edge adjacency matrix node with the weight value $w^{edge}_{i,j}$ of each respective edge. The computation complexity of electing an edge as a DS edge is $O(\Delta^2)$, as 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. ■

The second algorithm, namely MLEDS1 (Algorithm 2), is the first that computes a CEDS; it starts from an IEDS and connects it by adding edges that are bounded by DS nodes of the IEDS and 1-hop relay nodes of them (those with the largest *clPCI* index) who collectively cover their 2-hop neighborhood. Steps (1)–(14) are distributed and executed by each node u. Since adding edges in a distributed manner may result in redundant edge selection, MLEDS1 has a pruning phase (line 11). Line 15 is fictitious in order to fulfill the postcondition, i.e., it need not be run in practice.

Proposition 2: The computation complexity of MLEDS1 is $O(\Delta^3)$ in the worst case.

Proof: In order to connect the IEDS, each node u needs to check the status of its 1-hop neighbor set, which has a $O(\Delta)$ complexity. 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. ■

An improved version of the previous algorithm (MLEDS2) applies the more sophisticated pruning technique developed in [12] in order to reduce the size of the resulting connected edge dominating set. Due to space constraints, we omit its pseudocode and computational complexity here.

Finally, Algorithm 3 first creates a connected node dominating set and then computes a CEDS through the addition of edges. Note that for such a node dominating set, all nodes are within one-hop of a selected node, so if we can judiciously add such connecting edges (between selected and non-selected nodes), we will have a CEDS. Steps (1)–(18) are executed in a distributed fashion by every node u.

Proposition 3: The computation complexity of the relay node set election process is $O(\Delta^3)$.

Proof: The prioritization phase involves neighbor sorting based on *clPCI* value, which is a $O(\Delta * \log \Delta)$ operation. The worst case construction phase results when a node 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, node 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; this operation costs $O(\Delta^3)$. ■

Proposition 4: The computation complexity of the pruning phase is $O(\Delta^3)$.

Proof: A relay node u needs to check its 1-hop and 2-hop neighbors in order to decide if it will act as a relay node or not. 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. ■

Proposition 5: The computation complexity of transforming the MCDS to MCEDS is $O(\Delta^4)$ in the worst case.

Proof: The worst case computation complexity of the transformation process of the MCDS to MCEDS is when a non-DS node u has Δ non-DS neighbors and each one of them has Δ neighbors too. In such case node u needs to compare its 1-hop with Δ neighbors in the worst case, and the neighbor set comparison has a $O(\Delta)$ complexity. ■

B. On the Size Relationship Between IEDS and CEDS

Here we establish the relationship between the cardinality of an IEDS and the cardinality of its corresponding[1] CEDS.[2]

Theorem 2: Any IEDS of size $|IEDS|$ can be turned into a CEDS by adding $2 \times |IEDS|$ additional edges to the IEDS in the worst case.

Proof: We provide the proof sketch. Firstly, we will state a corollary that results immediately from the independent edge domination property, and then we will define the concept of *neighboring dominators* of an edge dominator e_v.

Corollary 1: In any IEDS, the closest (in terms of hops) edge dominator to any edge dominator can be found one or two hops away, i.e., ≤ 2 other edges are located in between these two edge dominators.

[1]That is, when $IEDS \subset CEDS$.

[2]Note that the claims of the theorem do not imply the relationship between the cardinality of the IEDS and that of the graph's edge set.

Algorithm 3: MLEDS#3

postcondition: Completed MCEDS election process

1 Identification of 1-hop ($N(u)$) and 2-hop ($N^2(u)$) neighborhood via distributed beaconing and calculation of clPCI indexes of the nodes;

2 **repeat**

3 Add in $R(u)$ a node $l \in N(u)$ with the largest *clPCI* index that covers at least one new node in $N^2(u)$;

4 **until** *each node in* $N^2(u)$ *is covered by node(s) in* $R(u)$

5 Announce $R(u)$;

6 **if** *selected as a relay node* **then**

7 $M(u) = T$; Announce status change;

8 Build $S^{constrained}_{(u)} = u_1, u_2, \ldots, u_n \mid u_{k\ (1 \leq k \leq n)} \in N(u) \wedge N^2(u)$, $M(u_{k\ (1 \leq k \leq n)}) = T$, $clPCI(u) < clPCI(u_{k\ (1 \leq k \leq n)})$;

9 **if** $S^{constrained}$ *is subject to* $N(u) \subset N(u_1) \cup N(u_2) \ldots \cup N(u_n)$ *and* $u_1, u_2, \ldots, u_n$ *form a connected graph* **then**

10 $M(u) = F$; Set M($w^{edge}_{i,j}$) = F any edge $w^{edge}_{i,j}$ incident to node u; `/* CDS Pruning */`

11 Announce status change; Return;

12 **end**

13 Build $S^{edge}_{(u)} = w^{edge}_{u,l_1}, w^{edge}_{u,l_2}, \ldots\ w^{edge}_{u,l_m} \mid w^{edge}_{u,l_k\ (1 \leq k \leq m)} \in$ E, $l_k \in$ N(u), $M(l_k) =$ F;

14 **if** $\exists\ w^{edge}_{u,l_k\ (1 \leq k \leq m)} \in S^{edge}_{(u)}$ *adjacent to a non DS edge and that edge is not incident to a DS node* **then**

15 M($w^{edge}_{u,l_k\ (1 \leq k \leq m)}$)=T; `/* MCDS to MCEDS */`

16 Announce status change;

17 **end**

18 **end**

19 Collect all edges (across the network) with a status=T;

Definition 9: A *neighboring edge dominator* e_u of an edge dominator e_v is any edge dominator which is at most two hops away from e_v.

An edge dominator e_v can have more than one neighboring edge dominator, but the exact number depends on network topology. Together, Corollary 1 and Definition 9 mean the topology between an edge dominator and its neighboring edge dominators must be one of the following:

- C1 An edge dominator has at least one neighboring edge dominator one hop away (e.g., edge dominator $\langle 1, 2\rangle$ is one hop away from $\langle 7, 9\rangle$ in Fig. 3).
- C2 An edge dominator has at least one neighboring dominator two hops away, and no dominators in one hop distance (edge dominator $\langle 1, 2\rangle$ from $\langle 4, 5\rangle$ in Fig. 3(Left)).

If [C1] holds for some dominator e_v, then we need to include one more edge dominatee into the EDS in order to connect e_v to its nearer neighboring dominator. If [C2] holds for some dominator e_v, then we need to include two more edge dominatees into the EDS in order to connect e_v to its nearer neighboring dominator. Thus, in the worst case, for every edge dominator, we need to include two more edges into the EDS in order to make it a CEDS. The worst case occurs for IEDS's as shown in Fig. 3 (Right). ■

V. Numerical Results

We performed an evaluation of the algorithms in MATLAB. Since there is no prior work on our topic, we use as baseline algorithm (referred to as BASE) the very popular one proposed in [13] for node dominating sets, which we augment

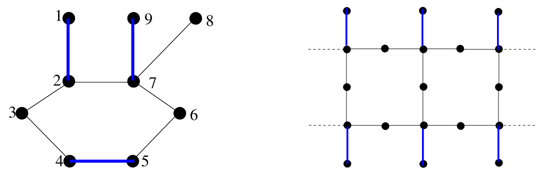

Fig. 3. (LEFT) An IEDS (blue thick edges) which exhibits all possible relative locations of neighboring edge dominators. For instance, edge dominator $\langle 1, 2\rangle$ is one hop away from $\langle 7, 9\rangle$ and two hops away from $\langle 4, 5\rangle$. (RIGHT) An IEDS (blue thick edges) which requires the maximum number of edge dominatees that must become dominators in order to get a CEDS. (Note that the graph extends infinitely to the left and to the right in the same pattern.)

TABLE I
COMPARISON OF PROPOSED ALGORITHMS TO A BASELINE ALGORITHM. FOR EACH COMPETITOR: THE LEFT COLUMN IS THE PERCENTAGE OF EDS SIZE W.R.T. NUMBER OF EDGES, AND THE RIGHT COLUMN IS THE PERCENTAGE OF INTERLAYER EDGES W.R.T. EDS SIZE

degree vs. (EDS size, # interlayer links)										
deg	MLEDS# 1		MLEDS# 2		MLEDS# 3		BASE		IEDS	
3	0.54	0.66	0.34	0.45	0.29	0.42	0.58	0.39	0.14	0.19
6	0.34	0.61	0.23	0.51	0.21	0.53	0.42	0.25	0.11	0.21
10	0.19	0.37	0.15	0.46	0.15	0.49	0.26	0.21	0.07	0.22
15	0.11	0.25	0.11	0.47	0.11	0.49	0.15	0.18	0.05	0.21
20	0.09	0.21	0.08	0.48	0.08	0.53	0.12	0.12	0.04	0.23
diameter vs. (EDS size, # interlayer links)										
diam	MLEDS# 1		MLEDS# 2		MLEDS# 3		BASE		IEDS	
3	0.21	0.39	0.21	0.48	0.16	0.51	0.25	0.29	0.08	0.21
5	0.32	0.54	0.32	0.47	0.20	0.50	0.36	0.33	0.10	0.20
8	0.33	0.57	0.33	0.45	0.21	0.47	0.39	0.38	0.10	0.20
12	0.46	0.64	0.46	0.43	0.25	0.44	0.51	0.41	0.12	0.19
17	0.55	0.66	0.55	0.45	0.32	0.42	0.62	0.42	0.15	0.20

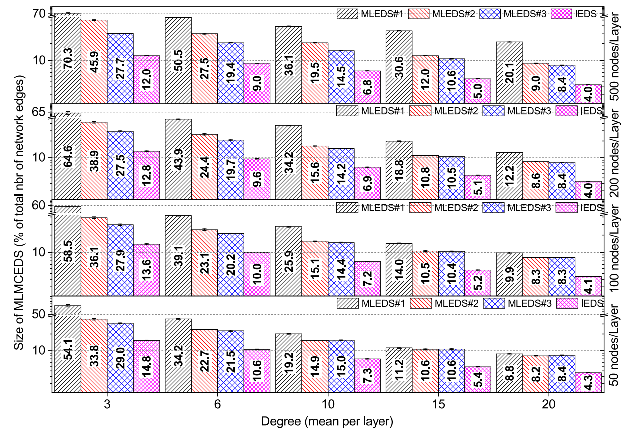

Fig. 4. Impact of the average node degree on the size of CEDS.

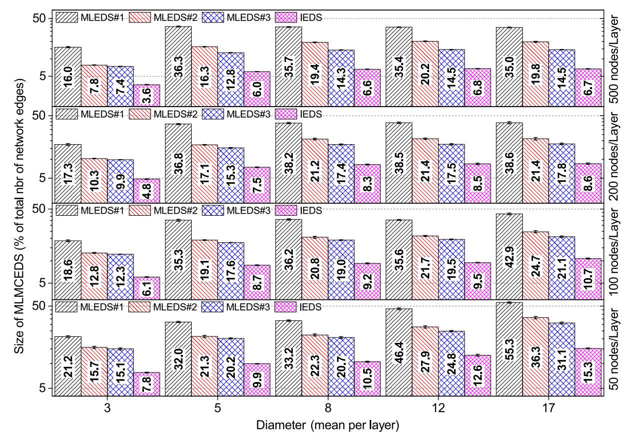

Fig. 5. Impact of the network diameter on the size of CEDS.

with a greedy heuristic to construct a connected EDS. We have also developed a generator [10] to produce multilayer networks. We use the size (in percentages) of the resulting (connected) EDS as the performance measure. The champion algorithm will be the one that calculates the smallest size CEDS. The default value for average node degree is set to 10, for network diameter it is set to 8, and for the number of layers it is set to 4. Each figure encompasses four sets of plots aligned vertically, corresponding to four different settings for the number of nodes in each of the layers.

In Table I we present the impact of average network degree and diameter on the competitors' EDS size for default settings, and also on the number of interlayer links included in the EDS as a resilience measure.

We can see that the proposed algorithms succeed in including many interlayer edges in the final CEDS; almost half of CEDS edges are interlayer ones. MLEDS3 in particular has stable behavior with respect to changes in network degree or diameter. On the other hand, BASE is the worst algorithm from the perspective of EDS size and this is consistent across all our experiments and therefore we refrain from presenting its performance in the sequel.

Fig. 4 shows the performance of the algorithms as the average degree varies between 3 and 20. The immediate observation is that when the degree increases, the size of the EDS decreases for all competitors, which is to be expected given that in dense topologies a single edge can dominate more edges. Also as expected are the observations that larger networks have relatively larger EDS's, as they must be sparser given that the average degree is fixed, and that the IEDS algorithm leads to the smallest EDS, as it does not have to ensure connectivity. Among the algorithms that created connected EDS, MLEDS3 is the best performing algorithm, creating an EDS twice the size of that calculated by IEDS which combined with Theorem 2 confirms that it is a good solution to our problem.

Fig. 5 shows the performance of the algorithms as the network diameter varies between 3 hops (so-called 'bushy' networks) to 17 hops ('long and skinny' topologies). As expected, in 'bushy' topologies, the resulting EDS's are smaller, whereas in the 'long and skinny' topologies more dominating edges are needed. As an analogy, in a star network (a 'bushy' topology) a single edge can dominate all others, whereas in a line topology with k connections, the connected edge dominating set has cardinality $k - 2$. Again, the best performing algorithm MLEDS3 is around 10% better than the second best algorithm on average. The performance gap reaches 25% for 'longer and skinnier' topologies.

Fig. 6 shows the performance of the algorithms as the number of layers varies. The increase in the number of layers causes the topology to become more connected, and

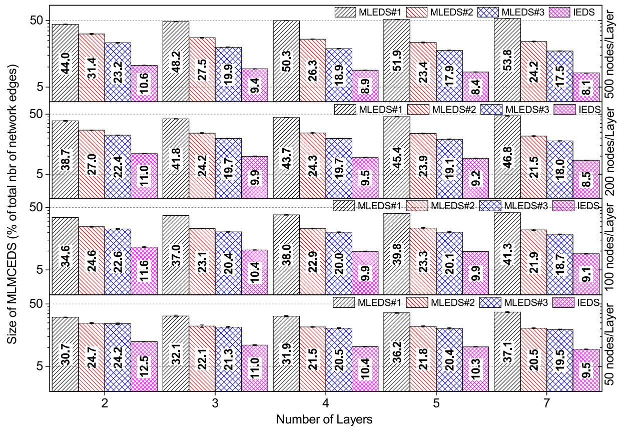

Fig. 6. Impact of the number of layers on the size of CEDS.

as a consequence the size of the EDS reduces, but not as dramatically as when the diameter shrinks or when the density increases. Again, MLEDS3 is the best performing algorithm.

VI. Related Work on MCMCEDS

The MCMCEDS problem, although novel per se, has connections to earlier work on finding minimum (connected) edge dominating sets. The MEDS problem has been shown to be NP-Complete in the single-layer case [14] in the centralized setting even for bipartite and planar graphs of maximum degree 3. Furthermore, even finding a $7/6-$approximation of the optimal set has been shown to be NP-Hard [15]. The MCEDS problem has also been shown to be NP-Complete [9, p. 102, Lemma 4.4.3].

MCMCEDS generalizes the plain (without any colors and any weights) EDS problem [14], if we assume that all edges have the same color. However, MCMCEDS cannot be transformed into the plain MCEDS problem with weights on edges [16] by assigning a uniform small weight to all inter-layer edges, and a uniform large one to all intra-layer edges, as in this case we might end up including *all* inter-layer edges into the dominating set simultaneously, which is not necessarily the most efficient solution. This is significant; while a $3 + \epsilon$ approximation exists for the weighted MCEDS problem [7], it will not apply to our MCMCEDS case. Problems related to *stratified domination* in graphs [17], [18] *ask for* a coloring of nodes, but in MCMCEDS, the colors are provided as part of the input to the problem. Problems related to *chromatic transversal domination* [19] are also not related to MCMCEDS for the same reason as stratified domination (in our case the colors are part of the input, and we do not seek a node coloring) and additionally because transversal domination demands that the dominating set's nodes should necessarily touch *all* color classes.

The problems most closely related to MCMCEDS are those reported in [5], where color classes are given, but domination is defined such that *all or none* of the graph elements (edges in our case) of a color class should be included in the dominating set. However, the MCMCEDS formulation allows for the inclusion of *any* number of edges belonging to *any* color class; therefore, the formulation is much more versatile (as compared to [5]) and encompasses a larger possible set of MCEDSs from which to choose from.

VII. Conclusion

Motivated by applications in traffic monitoring in diverse communication systems, we presented distributed algorithms for the creation of connected Multi-colored Edge Dominating Sets in multi-layer graphs. After showing that the underlying problem is hard to solve, we showed that a heuristic algorithm based on amending a connected node dominating set to create a connected edge dominating set provides the best performance. While our heuristics performed well over a range of scenarios, establishing approximability results for the MCMCEDS problem represents an important line of future work.

Acknowledgment

The authors would like to thank the anonymous reviewers for their comments, which have helped improve this letter.

References

[1] E. Bertino and N. Islam, "Botnets and Internet of Things security," *Computer*, vol. 50, no. 2, pp. 76–79, Feb. 2017.

[2] D. Papakostas, P. Basaras, D. Katsaros, and L. Tassiulas, "Backbone formation in military multi-layer ad hoc networks using complex network concepts," in *Proc. IEEE MILCOM*, 2016, pp. 842–848.

[3] T. W. Haynes, S. Hedetniemi, and P. Slater, *Fundamentals of Domination in Graphs* (Pure and Applied Mathematics). New York, NY, USA: Marcel Dekker, 1998.

[4] J. McCoy and M. A. Henning, "Locating and paired-dominating sets in graphs," *Discr. Appl. Math.*, vol. 157, no. 15, pp. 3268–3280, 2009.

[5] P. J. Slater, "Colored problems in graphs," *AKCE J. Graphs Combinatorics*, vol. 4, no. 2, pp. 191–196, 2007.

[6] G. Chartrand and M. J. Stewart, "The connectivity of line-graphs," *Mathematische Annalen*, vol. 182, no. 3, pp. 170–174, 1969.

[7] T. Fujito, "On approximability of the independent/connected edge dominating set problems," *Inf. Process. Lett.*, vol. 79, no. 6, pp. 261–266, 2001.

[8] J. Wu and H. Li, "On calculating connected dominating set for efficient routing in ad hoc wireless networks," in *Proc. ACM DIALM*, 1999, pp. 7–14.

[9] A. Munaro, "On some classical and new hypergraph invariants," Ph.D. dissertation, Dept. Informat. Appl. Math., Université Grenoble Alpes, Grenoble, France, 2016.

[10] P. Basaras, G. Iosifidis, D. Katsaros, and L. Tassiulas, "Identifying influential spreaders in complex multilayer networks: A centrality perspective," *IEEE Trans. Netw. Sci. Eng.*, to be published. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8114211/

[11] N. Dimokas, D. Katsaros, and Y. Manolopoulos, "Energy-efficient distributed clustering in wireless sensor networks," *J. Parallel Distrib. Comput.*, vol. 70, no. 4, pp. 371–383, 2010.

[12] D. Papakostas, S. Eshghi, D. Katsaros, and L. Tassiulas, "Energy-aware backbone formation in military multi-layer ad hoc networks," *Ad Hoc Netw.*, 2018. [Online]. Available: http://faculty.e-ce.uth.gr/dkatsar/adhocnetworks_multilayer.pdf

[13] M. Gerla and J.-C. Tsai, "Multicluster, mobile, multimedia radio network, *Wireless Netw.*, vol. 1, no. 3, pp. 255–265, 1995.

[14] M. Yannakakis and F. Gavril, "Edge dominating sets in graphs," *SIAM J. Appl. Math.*, vol. 38, no. 3, pp. 364–372, 1980.

[15] M. Chlebík and J. Chlebíková, "Approximation hardness of dominating set problems in bounded degree graphs," *Inf. Comput.*, vol. 206, no. 11, pp. 1264–1275, 2008.

[16] R. Carr, T. Fujito, G. Konjevod, and O. Parekh, "A $2\frac{1}{10}$-approximation algorithm for a generalization of the weighted edge-dominating set problem," *J. Comb. Optim.*, vol. 5, no. 3, pp. 317–326, 2001.

[17] G. J. Chang, C.-W. Chang, D. Kuo, and S.-H. Poon, "Algorithmic aspect of stratified domination in graphs," *Inf. Process. Lett.*, vol. 113, nos. 22–24, pp. 861–865, 2013.

[18] G. Chartrand, T. W. Haynes, M. A. Henning, and P. Zhang, "Stratification and domination in graphs," *Discr. Math.*, vol. 272, nos. 2–3, pp. 171–185, 2003.

[19] L. B. Michaelraj, S. K. Ayyaswamy, and S. Arumugam, "Chromatic transversal domination in graphs," *J. Comb. Math. Comb. Comput.*, vol. 75, pp. 33–40, Nov. 2010.