

# Backbone Formation in Military Multi-Layer Ad Hoc Networks Using Complex Network Concepts<sup>§</sup>

Dimitrios Papakostas\* and Pavlos Basaras\*<sup>†</sup>, Dimitrios Katsaros<sup>‡</sup>, Leandros Tassioulas<sup>‡</sup>

\*Department of Electrical & Computer Engineering, University of Thessaly, Greece

<sup>†</sup>CERTH, Greece

<sup>‡</sup>Department of Electrical Engineering & Yale Institute for Network Science, Yale University

**Abstract**—Modern battlefields are characterized by increasing deployment of ad hoc communications among allied entities. These networks can be seen as a complex multi-layer ad hoc network, where each layer may be an independently acting soldiers’ group, a group of drones, helicopters, vehicles and so on. Building a backbone network for these environments, which will guarantee efficient communication among all nodes (i.e., network-wide broadcasting) is of fundamental significance for the dissemination of information. In this article we generalize the concept of connected dominating sets for multi-layer networks and use them as the network backbone. We propose efficient methods to identify nodes that are efficient cross-layer spreaders along with a distributed algorithm to build the connected dominating set. Due to the lack of competing methods in the literature, we compare the proposed methods against some baseline methods and investigate the performance of all algorithms for a variety of multi-layer network topologies, illustrating their advantages and disadvantages; the result of the evaluation identifies the *cIPCI* method of recognizing efficient cross-layer spreaders as the champion method.

## 1. Introduction

Tactical ad hoc networks encompass some unique characteristics that differentiate them in terms of requirements, expectations, needs and constraints from the respective commercial. Those characteristics are related to dynamic topology, scarcity of bandwidth and excessive delay. Tactical wireless networks built with the Joint Tactical Radio System in mind have layers of subnets. These subnets are built up with waveforms (a waveform is a wireless multiple access radio frequency technology). 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.

All in all, we consider these ‘island’ subnetworks as being the layers of a single, large network, which we call a *multi-layer* communication network. Abstracting all the specifics, we illustrate such a multi-layer network in Figure 1 composed by a subnetwork (layer) of soldiers, a subnetwork (layer) of helicopters and a subnetwork (layer) of drones.

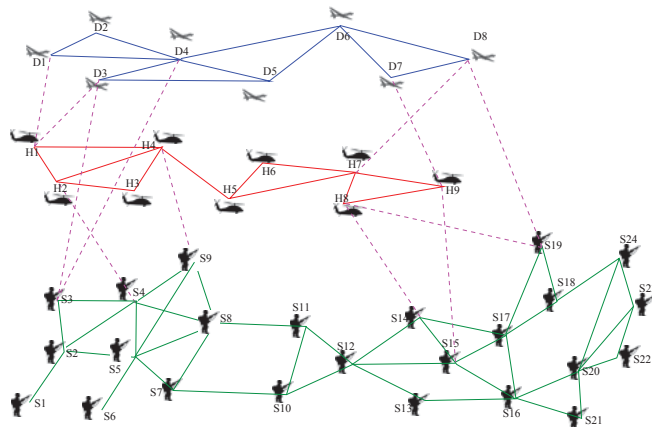


Figure 1: Abstraction of a multi-layer ad hoc network (for the purposes of illustration, physical obstacles have been removed, and the entities have been projected into the two dimensional space): The first layer is comprised of soldiers, the second of helicopters, and the third of drones. Solid (green/red/blue) color links denote communication links among entities of the same layer (soldiers/helicopters/drones). Dashed, purple links denote links among entities belonging to different layers.

Fast, information spreading across the whole network is vital to many battle and intelligence operations. Since each layer is an ad hoc wireless network, the goal is to construct a *backbone* connecting all these layers in such a way that efficient and effective information dissemination can take place. Among the most-researched methods for backbone formation in wireless ad hoc networks are those based on

<sup>§</sup>This work was partially supported by the US Office of Naval Research (ONR) under award N00014-14-1-2190. D. Katsaros’ work was done while he was on sabbatical leave at Yale university.

dominating sets [1] and those based on clustering [2]. Clustering can be problematic for modern battlefields because the participating units are highly mobile and thus frequent re-clustering might be necessary; this will jeopardize the communication ability of the entities and also put under question the robustness of the multi-layer network. Even the approaches for clustering ad hoc vehicular networks [3], [4] that take mobility into account will not work in a battlefield, because these approaches exploit the road network topology which is usually not present in a battlefield. On the other hand, backbones based on connected dominating sets [5] are a fine solution which combines flexibility (considerations for backbone diameter, different transmission ranges, interference, etc) and incorporation of even social-cognitive techniques [6].

The problem of constructing connected dominating set-based backbone for multi-layer ad hoc networks has not been considered in the literature so far, mainly because there is no supporting technology, but we envision its development in the near future, e.g., with software defined radio, Exelis SideHat. Even though traditional graph-theoretic concepts [5], [1] can be used for this problem, *network science* concepts such as centralities and communities detection can also provide essential tools for the management of military ad hoc networks [7], [8], [9], and especially in our case, they can help identify efficient cross-layer *dominators*. In this context, the article makes the following contributions:

- It introduces the problem of calculating minimum connected dominating sets in multilayer networks.
- It defines measures of *node importance* which help to identify those nodes ‘strategically’ positioned in the multi-layer network that will act as dominators.
- It develops a distributed algorithm for calculating the connected dominating set.
- It experimentally evaluates the proposed method for constructing connected dominating set against a robust competitor.

The rest of this article is organized as follows: section 2 provides background information, it explains why earlier methods for dominating set calculation can not work, and formally defines the investigated problem; section 3 presents the proposed measures and the distributed algorithm for dominating set construction; section 4 evaluates the performance of the proposed algorithm, section 5 presents related work, and section 6 concludes the article.

## 2. Backbone formation for multi-layer ad hoc networks

Firstly, we will provide some basic definitions on dominating sets [10] before we formulate the problem.

**Definition 1.** A dominating set  $DS(G)$  of a network  $(G, E)$  ( $G$  is the set of nodes, and  $E$  is the set of links among nodes) is any subset of  $G$  with the property that any node  $v$  of  $G$  is either a member of  $DS(G)$  (then,  $v$  is called a *dominator*) or  $v$  is one hop away from a dominator (then,  $v$  is called a *dominatee*).

**Definition 2.** A connected dominating set  $CDS(G)$  of a network  $(G, E)$  is any dominating set of  $G$  with the property that there is a path between any pair of dominators.

**Definition 3.** A minimum connected dominating set  $MCDS(G)$  of a network  $(G, E)$  is a  $CDS$  of  $G$  which is comprised of the minimum possible number of dominators.

It is easy to deduce that the MCDS for the drone-layer includes only the nodes  $\{D4, D6\}$ ; the MCDS for the helicopter-layers includes the nodes  $\{H4, H5, H7\}$ , and a (there exist more than one) MCDS for the soldier-layer includes the nodes  $\{S2, S5, S7, S10, S12, S15, S17, S16, S20\}$ . Finding the MCDS of a graph in the centralized setting (i.e., having knowledge of the complete network topology) belongs to the class of NP-complete problems [11]; it is understood that efficient (heuristic) distributed solutions to the same problem – which are those preferred for ad hoc networks – are much harder to devise [1].

**Definition 4.** A multi-layer network comprised of  $n$  layers is a pair  $(G^{ML}, E^{ML})$ , where  $G^{ML} = \{G^i, i = 1, \dots, n\}$  is a set of networks  $(G_i, E_i)$  as defined earlier, and a set of interlayer links  $E^{ML} = \{E_{i,j} \subseteq G_i \times G_j; i, j \in \{1, \dots, n\}, i \neq j\}$ .

In Figure 1,  $G_1 = \{S_i, i = 1, \dots, 24\}$ ,  $G_2 = \{H_i, i = 1, \dots, 9\}$ ,  $G_3 = \{D_i, i = 1, \dots, 8\}$ , and  $E^{ML}$  is the set of all links denoted by dashed lines, e.g.,  $(S_3, D_4)$ .

### 2.1. Problem formulation

Suppose that we are given an undirected (links are bidirectional), unweighted (no weights on links/vertices) network comprised of multiple (i.e., more than one) layers denoted as  $(G^{ML}, E^{ML})$ . Then, this article studies the **ML-MCDS** problem from a distributed perspective and it also develops a heuristic approximation to the **ML-MCDS** problem.

**Definition 5 (ML-MCDS problem).** Solve the Minimum-Connected Dominating Set for a multi-layer network in a distributed fashion, i.e., determine the set  $MCDS^{ML}$  comprised of the minimum number of nodes (belonging to any layer) such as: a) their *induced* subgraph is connected (with intra and/or inter-layer links) and the rest of the nodes (not belonging to  $MCDS^{ML}$ ) are adjacent to at least one node belonging to  $MCDS^{ML}$ , b) the number of dominators in each layer is the minimum one, c) having only knowledge of the  $k$ -hop neighborhood around each node. Here, we set  $k = 2$ .

Constraint (a) ensures connectivity of the backbone, and (c) enforces a distributed only approach. Constraint (b) needs some further discussion. We could have simply described it as ‘the total number of dominators is the minimum one’. It is obvious that such a formulation does not imply the one we have used in Definition 5, but the reverse is

true. Thus we have strived for a stronger formulation which can alleviate problems arising from multi-layer networks when their relative size (measured in number of nodes) is highly skewed. Therefore, our definition strives for locating efficient ‘cross-layer’ dominators.

It is easy to prove that our problem is NP-complete [11]. Apparently, solving the same problem for directed (i.e., unidirectional links) and/or weighted (energy considerations on links) versions of networks is also very interesting and subject to solutions not unlike the ones proposed here. Similarly, the problem of stability or incremental maintenance of a discovered ML-CDS (in cases of attacks to nodes, or due to nodes’ departures/moves) is also very significant [12], but for the interest of space will not be discussed here.

We admit that we have provided a completely abstract formulation of the problem without taking into account practical constraints/considerations such as types of military formations, physical obstacles and so no; these are reflected in an abstract way into the resulting network topology. Nevertheless, we feel that such considerations will certainly provide optimizations opportunities worth examining in a separate article.

## 2.2. Decomposition-based and aggregation-based approaches for DS calculation in multi-layer networks will not work

A method that calculates (in a centralized or a distributed fashion) a CDS for each layer separately – thus applying a *decomposition approach* – and then trying to connect them, is clearly a suboptimal solution. This approach is a characteristic case of the problem where we have calculated an unconnected dominating set of a network and we need to find a set of dominee nodes in order to connect the dominator nodes. In this case, as Theorem 1 tells us, the number of nodes that need to be added to the DS in order to become a CDS can be (in the worst case) equal to two times the size of the DS.

**Theorem 1.** Any (unconnected) dominating set of size  $|DS|$  can be turned into a Connected Dominating Set by adding  $2 \times |DS|$  additional nodes in the dominating set in the worst case.

*[Sketch of] Proof.*

Firstly, we will state a corollary that results immediately from the domination property, and then we will define the concept of *neighboring dominators* of a dominator  $v$ .

**Corollary 1.** In any dominating set, the closest (in terms of hops) dominator to any dominator can be found at one, two or three hops away, i.e., at most three hops away.

**Definition 6.** A *neighboring dominator*  $u$  of a dominator  $v$  is any dominator which is at most three hops away from  $v$ .

A dominator  $v$  can have more than one neighboring dominators, but the exact number depends on the network topology. Combining Corollary 1 and Definition 6, we can

recognize only three cases that describe the topology between a dominator and its neighboring dominators:

- C1 A dominator has at least one neighboring dominator one hop away (dominator S1 – and S7 of course – in Figure 2).
- C2 A dominator has at least one neighboring dominator two hops away, and none of the rest dominators in one hop distance away (dominator S17 in Figure 2).
- C3 A dominator has at least one neighboring dominator three hops away, and none of the rest dominators in one or two hops distance away (dominators S10 and S14 in Figure 2).

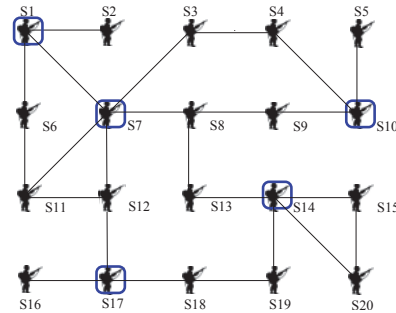


Figure 2: A dominating set (composed of blue nodes) which exhibits all possible relative locations of neighboring dominators.

If [C1] holds for each and every dominator, the DS is a CDS. If [C2] holds for some dominator  $v$ , then we need to include one more dominee into the DS in order to connect  $v$  to its nearer neighboring dominator. Finally, if [C3] holds for some dominator  $v$ , then we need to include two more dominees into the DS in order to connect  $v$  to its nearer neighboring dominator. Thus, in the worst case, for every dominator we need to include two more nodes in the DS in order to make it a CDS. The worst case occurs for dominating sets as that shown in Figure 3.  $\dashv$

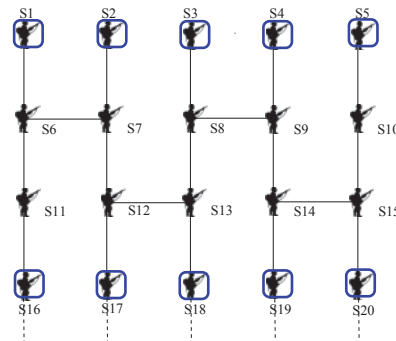


Figure 3: A dominating set (composed of blue nodes) which requires the maximum number of dominees that must become dominators in order that the resulting DS is a CDS.

Even though Theorem 1 applies in the worst case only, it is quite possible that the military formations in battlefields

will make topologies such as that of case C3 to appear quite frequently. Thus the decomposition-based approaches will create long-and-skinny CDS instead of ‘bushy’ ones, resulting in large communication latencies.

On the other hand, if we apply an *aggregation approach* treating all links the same even though some of them may connect nodes belonging to different layers, will cause other types of problems; looking again at Figure 1 and following an aggregation-based method, some algorithm might decide to include node S4 into the dominating set, because it is the most connected node in its neighborhood. However, a wiser decision would be to include node S3 into the dominating set, because that node connects to node D3 and D4, with the former providing links to the helicopter-level and the latter being the most connected to its level, thus better facilitating information dissemination across all layers.

Therefore, neither a decomposition- nor an aggregation-based approach would provide efficient solutions to our problem.

### 3. Identifying (efficient) cross-layer dominators

The discussion in the previous section highlighted the significance of assessing and exploiting a node’s intra- and inter-layer links in order to be considered as a candidate dominator. Assessing the significance of a node must be quick (in small computational complexity) and cheap (in small communication complexity, which practically implies small energy consumption as well). Therefore, we need to devise a method that will rely on connectivity information from the node’s ‘local’ neighborhood (one or two hops away at most), and without computing sophisticated functions for assessing the value of the node. For the case of single-layer networks, a node’s degree [13], [14] is a measure that complies with the above requirements, but has several drawbacks [1].

Our plain intuition for selecting nodes that will eventually be ‘efficient’ (i.e., they will cover as much of a network area as possible) dominators is that these nodes should be strategically located in dense areas of the (multi-layer) network. In [6] we showed how to identify such nodes in a single-layer network by defining the *Power Community Index (PCI)*, which is a node centrality measure.

**Definition 7 (Power Community Index (PCI) [6]).** The *PCI* index of a node  $v$  is equal to  $k$ , such that there are up to  $k$  nodes in the 1-hop neighborhood of  $v$  with degree greater than or equal to  $k$ , and the rest of the nodes in that 1-hop neighborhood have a degree less than or equal to  $k$ .

Now turning to our multi-layer network case, we can straightforwardly generalize it for multi-layer networks by ignoring(!) the existence of layers; then we get the *Layer-agnostic PCI (laPCI)* defined as follows:

**Definition 8 (Layer-agnostic PCI (laPCI)).** A node has *laPCI* equal to  $k$ , if it has  $k$  one-hop neighbors with a number of links *towards any* layer greater than or equal

to  $k$ , and the rest of its one-hop neighbors have a number of links towards any layer less than or equal to  $k$ .

*laPCI* gives credit to a node whose neighbors have many connections in different layers, however, it makes no distinction on how those connections are distributed over the layers, which is problematic. We can cure this, by taking into account the existence of layers:

**Definition 9 (Minimal-layers PCI ( $mlPCI_n$ )).** A node has *mlPCI<sub>n</sub>* equal to  $k$ , if it has  $k$  one-hop neighbors with the number of links *towards at least  $n$*  layers greater than or equal to  $k$ , and the rest of its one-hop neighbors have a number of links toward at least  $n$  layers less than or equal to  $k$ .

*mlPCI<sub>n</sub>* characterizes a node for its connectivity in a predefined number of layers. We further combine *mlPCI<sub>n</sub>* values for all  $n$  bringing *mlPCI* for a node  $v$  in its final form:

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

*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 *laPCI* and *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 7) 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)*. We calculate *xPCI* for node *S4* in Figure 1 by observing that only nodes *S5*, *S8* and *S9* are responsible for defining *PCI(S4)* at the ‘Soldiers’ layer and node *H2* at the ‘Helicopters’ layer, thus  $xPCI(S4) = 5$ . *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 the *xPCI* 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 to obtain reasonable numbers even for large networks). We call this new measure *Cross-layer PCI (clPCI)*.

#### 3.1. Distributed CDS in multi-layer networks

Here, we describe a distributed CDS generation protocol which makes use of any of the proposed measures (for illustration purposes, we use *clPCI* in the pseudocode).

##### Gathering Data:

- Nodes via the exchange of “Hello” messages gather their 1-hop ( $N(u)$ ) and 2-hop ( $N^2(u)$ ) neighborhood connectivity.



- Each node calculates and broadcasts its  $cIPCI$  index. Hence, each node  $u$  is aware of the  $cIPCI$  values in  $N(u)$ .

#### Node Selection:

- For any node  $u$ , nodes in  $N(u)$  are sorted in decreasing order of their  $cIPCI$  values.
- Since the algorithm is executed in a distributed fashion, node  $u$  first selects as its relays those 1-hop neighbors that have already been selected as dominators by other neighboring nodes (if any).
- While there are still nodes in  $N^2(u)$  which are not neighbors to any node in the set of  $u$ 's relays, select and include in the set of  $u$ 's relays the next node from  $N(u)$  with the largest  $cIPCI$  index that covers at least one new node in  $N^2(u)$

It is easy to prove the correctness (i.e., it computes a CDS) of the algorithm. Clearly, the computation complexity of this algorithm is dominated by the sorting process –  $O(m \times \log_2 m)$ , for a node with where  $m$  neighbors for the computation of the measures. The communication complexity (per node) is constant (only 2 messages), i.e.,  $O(n)$  for a  $n$ -node network.

## 4. Experimental evaluation

We performed a simulation-based performance evaluation of the proposed methods in MATLAB.

### 4.1. Experimental settings

**4.1.1. Competitors.** As mentioned in the introduction, there is no prior work on our topic; therefore we used as baseline competitors source-initiated versions of the degree [13] and OLSR [14] in the ‘aggregated’ complex network where all layers have been collapsed into a single one. We also include source-initiated versions of all the proposed ones, namely  $laPCI$ ,  $mlPCI$  and  $cIPCI$ .

**4.1.2. Performance measures.** We use the size of the resulting connected dominating set as the measure that quantifies the performance of the competing algorithms in Figures 4 and 5, and the same measure per layer in Figure 6. Apparently, a small CDS implies small energy consumption, and in most cases it also implies a short latency; the latency depends also of the ‘shape’ of the dominating set (long versus bushy ones).

**4.1.3. Datasets.** Due to the lack of publicly available, real-world military multi-layer networks, we developed a generator for multi-layer networks in MATLAB. Our aim was to build a generator that could generate in an algorithmic way a variety of multi-layer network topologies, so as to be able to explain the obtained results afterwards with respect to the topology. The generator was developed and described in detail in [15], but here, we will present its basic features.

parameter	range	default
avg. node degree (D)	6, 8, 10, 12, 16	10
network diameter (H)	7, 15, 30, 40	7
# network layers (L)	2, 3, 4, 5, 6, 7	5
topology skewness	0.1/0.1/0.1 0.1/0.9/0.1 0.1/0.5/0.5 0.5/0.5/0.1 0.9/0.5/0.1	0.1/0.5/0.5
relative size of one layer to its previous	10%, 20%, 40%, 70%	40%

TABLE 1: Experimentation parameters values.

We used unit disk random graphs (UDG) [16] controlled by three independent parameters: firstly, the link density in each layer which is expressed by the average degree  $D$  of each node, secondly the number of nodes per layer (i.e., size of the layer), and the number of layers ( $L$ ). Non-uniform intra-layer models could also be used to better approach reality. 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 may require ‘coverage’ (preference) for a certain layer, that is, nodes generating most of their interconnections towards a specific layer, e.g., most interlinks from the drones layer are generated towards the soldiers network, etc. With the above considerations we apply the Zipfian distribution for our interconnectivity generator. The desired skewness is managed by parameter  $s \in (0, 1)$ . We apply three distinct Zipfian laws, 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.

In the next section we will represent the values of these parameters (which collectively will be called *topology skewness*) as a sequence of three floats, e.g., 0.5/0.5/0.1, meaning that  $s_{degree} = 0.5$ ,  $s_{layer} = 0.5$  and  $s_{node} = 0.1$ . Finally, in a multilayer network the relative size of the layers would clearly have an impact on the performance of the algorithms. Thus, we equipped our topology generator with the ability to create multi-layer topologies where each can be a percentage (10%, 20%, 40%, 70%) larger than the previous one. So we may 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.

### 4.2. Experimental results

**4.2.1. Impact of topology density.** Firstly, we wish to evaluate the impact of topology density on the performance of the algorithms. To keep the experiment controlled, we

vary the density of a single layer keeping the rest unaltered to the extent possible. The results are illustrated in the bottom plot of Figure 4.

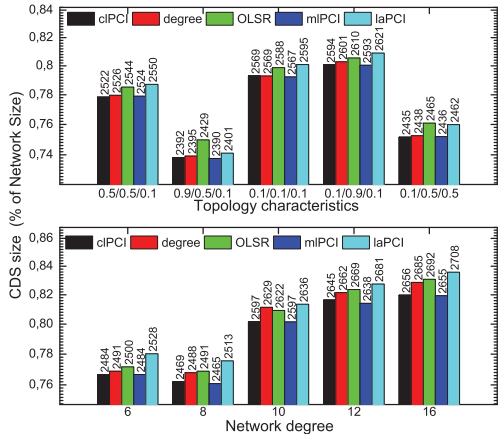


Figure 4: Impact of network density and topology skewness on the size of CDS.

Our first observation concerns the size of the generated *CDS* formed by each competitor; one would expect that a higher network density would decrease *CDS* size, but here we observe that *CDS* size increases. This is due to the fact that the inter-layer links are spread more uniformly among layers, and thus there are no ‘hub nodes’ whose existence will result in a decreasing *CDS* size for increasing density. Concerning the performance of the competitors, *clPCI* and *mlPCI* produce the smallest *CDS*s for all  $D$  values, but none of them are better than the other. Moreover, their performance gap from the third best performing algorithms widens with increasing density, which is due to the fact they exploit the inter-layer links to identify cross-layer dominators. Now looking at the upper part of Figure 4, where the champion algorithms remain the same as before, we observe the generic trend that *CDS* size for all competitors increases when the topology skew is small, i.e.,  $s_{degree}$  and  $s_{node}$  have small values. Only, when there are ‘hub nodes’ i.e.,  $s_{degree} = 0.9$  the size of *CDS* is small.

**4.2.2. Impact of network diameter.** In Figure 5 we evaluate the effect of the multi-layer network diameter in the size of the *CDS*. At this point we need to say, the each layer is composed of around 500 nodes, and  $n$ -layered network is composed of the previous  $n - 1$  layers plus one more layer. As the network diameter increases the size of the constructed *CDS* for all methods decreases. The decrement of the diameter is the result of sparser vicinities, i.e., fewer links between the network nodes. In other words, fewer, longer (in hops), and more distinct paths towards the nodes of the multilayer network, which renders the election of those nodes that cover the  $N^2$  neighborhood more discrete, and hence fewer nodes are recruited. Focusing on the evaluation of the competitors, we observe that the difference in their performance is minimum when  $H = 7$ . This is due to fact that when nodes are relatively “close” to each other, there is significant overlapping in the selected *CDS*s.

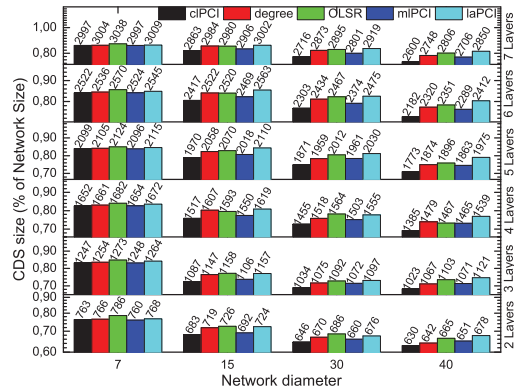


Figure 5: Impact of network diameter on the *CDS* size.

**4.2.3. Impact of increasing the layer size.** Figure 6 illustrates the impact of the number and size of layers on *CDS* size; as expected, the generic trend is that the size of *CDS* increases with more layers or more variability in the relative layer size. We need to say here, that the top layer (Layer 5) in each 5-layered network is composed of 500 nodes and the remaining layers have increased size with respect to the previous layer as depicted in the  $x$ -axis. The purpose of Figure 6 is to clarify that the performance of the proposed methods is the result of selecting a minimum dominating set in each respective layer, which due to a careful selection of key intra & interconnected nodes, results in an interconnected dominating set, i.e., an  $MCDS^{ML}$ . Evidently, as the size of each layer increases, so does the cardinality of the elected *CDS*s for all methods. The competitors’s ranking obtained from the previous subsection has remained unchanged, i.e., *clPCI* selects the smallest *CDS* in all layers, and thus the overall minimum  $MCDS^{ML}$ , which highlights the effectiveness of the proposed technique.

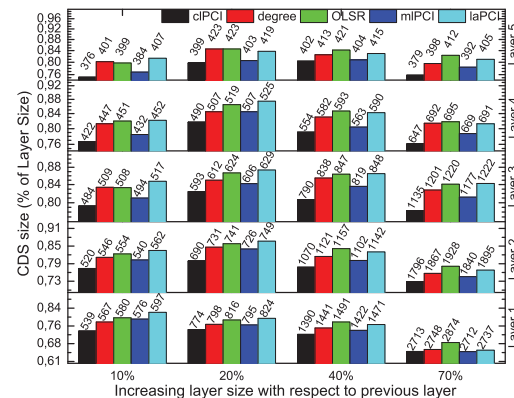


Figure 6: Impact of network size on the *CDS* size.

## 5. Related work

We have already stated in section 1 that there is no previous work on calculating connected dominating sets in multi-layer complex networks. Nevertheless, the topic of this article is relevant to a number of areas which we briefly discuss here. The wireless ad hoc networks com-

munity has extensively investigated the topic of distributed algorithms for (single layer) ad hoc networks. Nice surveys of recent results on this topic are presented in [1], [17]. The main focus of these works is to produce a small connected dominating set under various constraints, such as type (unidirectional/bidirectional) of links, backbone diameter length, energy budget, and so on. Research on multi-layer complex networks [18], [19] (generalization of multiplex, of interdependent networks) is relatively new and spans directions such as formation mechanisms [20], centralities, communities [21], diffusion processes [22]. The algorithms developed in this article are more close to the concept of influential spreaders. Influential spreaders in a complex networks are those nodes which under a specific spreading model (e.g., SIR, SIS) are able to spread the ‘infection’ in a large part of the network. After the seminal work of [23], measures such as *k-shell* [23], *pci* [24] and others have been proposed to identify influential spreaders over single-layer complex networks. There is some work on positive influence dominating sets in single layer social networks [25]. Finally, our work [15] investigated the issue of detecting influential spreaders for multi-layer complex networks using concepts similar to those presented here.

## 6. Conclusions

We considered the problem of backbone formation for modern military ad hoc networks which are composed by multiple subnetworks (‘layers’ in this article’s terminology). We investigated the possibility of forming the backbone in terms of connected dominating sets, and subsequently we defined – for the first time in the literature – the problem of minimum connected dominating set for multi-layer networks. We recognized the significance of determining ‘efficient’ cross-layer dominators, and proposed a set of measures (based on network theory concepts) for detecting them, namely *laPCI*, *mPCI* and *clPCI*. Then, we proposed a distribution algorithm for backbone formation based on those measures. We performed a simulation-based evaluation of the proposed techniques against baseline methods (i.e., degree, OLSR) and showed that the distributed algorithm based on *clPCI* shows (almost always) the best performance.

## References

- [1] J. Yu, N. Wang, G. Wang, and D. Yu, “Connected dominating sets in wireless ad hoc and sensor networks: A comprehensive survey,” *Computer Communications*, vol. 24, no. 6, pp. 121–134, 2013.
- [2] O. Younis, M. Krunz, and S. Ramasubramanian, “Node clustering in wireless sensor networks: Recent developments and deployment challenges,” *IEEE Network magazine*, vol. 20, no. 3, pp. 20–25, 2006.
- [3] L. Maglaras and D. Katsaros, “Social clustering of vehicles based on semi-Markov processes,” *IEEE Transactions on Vehicular Technology*, vol. 65, no. 1, pp. 318–332, 2016.
- [4] I. Rubin, A. Baiocchi, F. Cuomo, and P. Salvo, “Vehicular backbone network approach to vehicular military ad hoc networks,” in *Proceedings of IEEE MILCOM*, 2013, pp. 901–909.
- [5] D.-Z. Du and P.-J. Wan, *Connected Dominating Set: Theory and Applications*. Springer, 2013, vol. 77.
- [6] N. Dimokas, D. Katsaros, and Y. Manolopoulos, “High performance, low complexity cooperative caching for wireless sensor networks,” *ACM/Springer Wireless Networks*, vol. 17, no. 3, pp. 717–737, 2011.
- [7] C. He, H. Ma, S. Kang, and R. Cui, “An overlapping community detection algorithm based on link clustering in complex networks,” in *Proceedings of IEEE MILCOM*, 2014, pp. 865–870.
- [8] J. P. Macker and D. J. Claypool, “Dynamic communities in evolving network graphs,” in *Proceedings of IEEE MILCOM*, 2012.
- [9] J. P. Macker and I. J. Taylor, “Prediction and planning of distributed task management using network centrality,” in *Proceedings of IEEE MILCOM*, 2014, pp. 859–864.
- [10] T. W. Haynes, S. Hedetniemi, and P. Slater, *Fundamentals of Domination in Graphs*, ser. Chapman & Hall/CRC Pure and Applied Mathematics. CRC Press, 1998.
- [11] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979.
- [12] J. P. Macker, B. Adamson, and D. J. Claypool, “Temporal stability for dynamic network relay sets,” in *Proceedings of IEEE MILCOM*, 2012.
- [13] M. Gerla and J.-C. Tsai, “Multicluster, mobile, multimedia radio network,” *ACM/Springer Wireless Networks*, vol. 1, no. 3, pp. 255–265, 1995.
- [14] A. Qayyum, L. Viennot, and A. Laouiti, “Multipoint relaying: An efficient technique for flooding in mobile wireless networks,” INRIA, Tech. Rep. 3898, March 2000.
- [15] P. Basaras, G. Iosifidis, D. Katsaros, and L. Tassiulas, “Identifying influential spreaders in multi-layer complex networks,” 2016, submitted.
- [16] F. A. Onat and I. Stojmenovic, “Generating random graphs for wireless actuator networks,” in *Proceedings of IEEE WoWMoM*, 2007.
- [17] P. Ruiz and P. Bouvry, “Survey on broadcast algorithms for mobile ad hoc networks,” *ACM Computing Surveys*, vol. 48, no. 1, pp. 1–35, 2015.
- [18] S. Boccaletti, G. Bianconi, R. Criado, C. I. del Genio, J. Gomez Gardenes, M. Romance, I. Sendina-Nadal, Z. Wang, and M. Zanin, “The structure and dynamics of multilayer networks,” *Physics Reports*, vol. 544, pp. 1–122, 2014.
- [19] K. Mikko, A. Arenas, M. Barthelemy, J. P. Gleeson, Y. Moreno, and M. A. Porter, “Multilayer networks,” *Journal of Complex Networks*, vol. 2, no. 3, pp. 203–271, 2014.
- [20] E. M. Shahrivar and S. Sundaram, “The strategic formation of multi-layer networks,” *IEEE Transactions on Network Science and Engineering*, vol. 2, no. 4, pp. 164–178, 2015.
- [21] L. G. S. Jeub, M. W. Mahoney, P. J. Mucha, and M. A. Porter, “A local perspective on community structure in multilayer networks,” *Network Science*, 2015, under review.
- [22] M. M. Salehi, R. Sharma, M. Marzolla, M. Magnani, P. Siyari, and D. Montesi, “Spreading processes in multilayer networks,” *IEEE Transactions on Network Science and Engineering*, vol. 2, no. 2, pp. 65–83, 2015.
- [23] M. Kitsak, L. K. Gallos, S. Havlin, F. Liljeros, L. Muchnik, H. E. Stanley, and H. A. Makse, “Identification of influential spreaders in complex networks,” *Nature Physics*, vol. 6, pp. 888–893, 2010.
- [24] P. Basaras, D. Katsaros, and T. L., “Detecting influential spreaders in complex, dynamic networks,” *IEEE Computer magazine*, vol. 46, no. 4, pp. 26–31, 2013.
- [25] F. Wang, H. Du, E. Camacho, K. Xu, W. Lee, Y. Shi, and S. Shan, “On positive influence dominating sets in social networks,” *Theoretical Computer Science*, vol. 412, no. 3, pp. 265–269, 2011.