2021 16th Annual Conference on Wireless On-demand Network Systems and Services (WONS)

Backbones for Internet of Battlefield Things

Dimitrios Papakostas, Theodoros Kasidakis, Evangelia Fragkou, Dimitrios Katsaros
Department of Electrical & Computer Engineering
University of Thessaly, Greece
{papdimit, tkasidakis, efragkou, dkatsar}@e-ce.uth.gr

***Abstract*—The Internet of Battlefield Things is a relatively new cyberphysical system and even though it shares a lot of concepts from the Internet of Things and wireless ad hoc networking in general, a lot of research is required to address its scale and peculiarities. In this article we examine a fundamental problem pertaining to the routing/dissemination of information, namely the construction of a backbone. We model an IoBT ad hoc network as a multilayer network and employ the concept of domination for multilayer networks which is a complete departure from the volume of earlier works, in order to select sets of nodes that will support the routing of information. Even though there is huge literature on similar topics during the past many years, the problem in military (IoBT) networks is quite different since these wireless networks are multilayer networks and treating them as a single (flat) network or treating each layer in isolation and calculating dominating set produces submoptimal or bad solutions; thus all the past literature which deals with single layer (flat) networks is in principle inappropriate. We design a new, distributed algorithm for calculating connected dominating sets which produces dominating sets of small cardinality. We evaluate the proposed algorithm on synthetic topologies, and compare it against the only two existing competitors. The proposed algorithm establishes itself as the clear winner in all experiments.**

***Index Terms*—Dominating sets, multilayer networks, Internet of Battlefield Things, adhoc networking**

I. INTRODUCTION

During the last decade the Internet of Things (IoT) took major steps towards its realization due to the abundance of wireless networks, the extreme miniaturization of devices, their supply with significant computing, communication and control power, and the further development of algorithmic solutions in distributed computing.

As expected, the advances in IoT inevitably impacted upon modern military operations, and specifically upon the modern and future battlefield which is populated by tens of thousands of "things", such as humans, vehicles, unmanned aerial vehicles (UAVs), aircrafts, sensors, etc. performing a wide variety of tasks including sensing the environment, communicating, acting in isolation and in cooperation [15], [24]. Thus, a new term and a new development/study area was born, namely the Internet of Battle(field) Things [15] or the Internet of Military Things (IoMT)[1] whose high level goal is to create a network of communication with any kind of "device". These devices are dynamically connected to meet multiple and often diverse missions, operate in a (semi)autonomic mode, and execute battlefield operations in order to support end-to-end control and command. A bulleted description of the significant aspects of IoBT from the perspective of a US Army research program is quite comprehensive[2].

Even though IoBT stems from IoT, a number of challenging characteristics [2] distinguish it from traditional IoT. We briefly describe the most significant features below:

- *Diversity in tasks and goals*. There will likely be many networks operating at the same time to achieve their particular goal, e.g., surveillance, tracking, attack.
- *Operation in dynamic and resource starving environments*. Some "devices" comprising the IoBT network might be energy-starving such as sensors, drones, others might not have energy issues, but they might be obliged to travel in not well chartered territories (e.g., planes).
- *Extreme device heterogeneity*. An IoBT network might include from tiny little sensors to some as big as armoured fighting vehicles.
- *High variance in network size and density*. An IoBT might be comprised for instance by the highly dense and cluttered network of a drone swarm, or it might be comprised by the union of a network formed by the soldiers of a battalion spread over a large area and the network formed by a tank platoon.

One of the major challenges for IoBT comprises the so-called *assured synthesis* and in particular the tasks of *recruitment* and *network composition* [2]. The former deals with the discovery of cyberphysical assets, the human assets and their particularities, and finally, with the the resilience to adversarial behaviour. The latter deals with the issue of dictating the set of nodes that must be included that satisfy the requirements and constraints of the planned mission.

A. Motivation and contributions

We took a first step [19] in dealing with the extreme heterogeneity of such a network by modelling it as a *multilayer network* [7]. We showed analytically that treating such a network as a plain union of independent subnetworks does not provide efficient solutions, at least for the task of facilitating fast packet/information routing. So, even though there is huge literature during the past many years, on dominating sets for

---

[1] https://www.computer.org/publications/tech-news/research/internet-of-military-battlefield-things-iomt -iobt

[2] https://www.arl.army.mil/business/collaborative-alliances/current-cras/iobt-cra/

ISBN 978-3-903176-35-5 © IFIP

single layer ad hoc networks, the problem in military (IoBT) networks is quite different since these wireless networks are multilayer networks and treating them as a single (flat) network or treating each layer in isolation and calculating dominating set produces submoptimal or bad solutions; thus all the past literature which deals with single layer (flat) networks is in principle inappropriate. In subsequent works, we developed distributed algorithms for monitoring and control of the communication links to deal with attacks [20], [21], and also with fact-finding algorithms to account for the social sensing aspects, e.g., characterize human assets/sources [14]. Moreover, in [19] we developed low-communication cost, distributed algorithms for selecting a set of nodes to comprise a backbone for a multilayer network – in our case, the IoBT network, based on the concept of node domination. Independently of our efforts and in the field of biological sciences, the work in [16] developed (centralized) algorithms for calculating dominating sets for multilayer networks.

However, neither the algorithms developed in [16] nor those developed in [19] can adequately serve the purpose we are seeking in this article. The former because it is centralized and thus can not work in the realm of IoBT, and moreover because it produces unconnected network spanners, and thus it can not guarantee free flow of information. The main issue with the latter algorithms is the fact that they produce not very compact spanner, i.e., there is still room for improvements as far as the size of the produced connected dominating set is concerned. This is of crucial significance, should we wish to have solutions scalable for enormous IoBT networks.

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

- It presents a new distributed algorithm $CCDS$ for calculating connected dominating sets for IoBT ad hoc networks, by applying an efficient pruning mechanism to reduce the size of the dominating set.
- It enhances $FAST-MDSM$ [16] so as to produce connected dominating sets for multilayer networks, i.e., we get a new algorithm called $FAST-CMDSM$.
- It implements the only existing competitors, namely the algorithm in [19] and $FAST-CMDSM$.
- It compares exhaustively the new algorithm against the aforementioned competitors using synthetically generated network topologies, and shows their shortcomings and advantages. The evaluation attests the overall superiority of the proposed algorithm.

The rest of this article is organized as follows: Section II describes the related work on the article's topic; Section III formulates the problem in mathematical terms, and Section IV develops distributed solutions for it. Section V evaluates the competing algorithms, and finally, Section VI summarizes the article.

II. RELATED WORK

The concept of Internet of Battle(field) Things emerged during the last five years or so [1], [15]. Since then, intense research is being conducted in issues such as designing backbones for routing [19], designing secure and reconfigurable IoBT networks [10], supporting secure information exchange [27], performing enemy localization [11], conducting communication link control [20], combating attacks at nodes [3], malware and fake news detection [4], [5], [14], and human assets protection [8], [17].

The most closely related work to the present article are those that deal with backbones that support routing in ad hoc type of wireless networks. Even though there is huge literature on this topic during the past twenty-five years [25], [29], [30], the problem in military (IoBT) networks is quite different since these wireless networks are modelled as multilayer networks [10], [19] (see Figure 1), and thus all the past literature which deals with single layer networks is in principle inappropriate. As mentioned in [19, Theorem 1], such approaches can produce suboptimal solutions meaning that it is better to announce as a dominator a node with a few interlayer links rather than one with many intralayer links.

The first approach in designing a backbone for military multilayer networks is reported in [19], where a distributed backbone formation algorithm named $clPCI$ was presented there. That algorithm utilized the well-known concept of (connected) dominating set (CDS) for forming the backbone. The choice of dominating set concept was made because the battlefield requires a) distributed algorithms: even though the calculation of a minimum (in size) CDS is in NP-Complete, there exist efficient distributed algorithms for this problem, b) resilient solutions: indeed there is mature knowledge on how to create a multi-connected, multi-dominating CDS, c) energy-aware solutions (e.g., sleep scheduling): there is also mature theory and practice on how to create e.g., multiple dominating sets, i.e., the so-called domatic partitions [23]. So, $clPCI$ was the first algorithm to construct a connected dominating set in a distributed way for multilayer networks. Until then, there was no prior work on the topic of calculating (connected) dominating sets for multilayer networks, probably because it was mistakenly conceived that that a multilayer network is equivalent (from the perspective of computing CDS) to a single layer network after ignoring layer information.

In biological sciences, where the concept of multilayer networks has been successfully used to model interacting networks, the algorithm $FAST-MDSM$ was presented at [16] for calculating dominating sets in multilayer networks. $FAST-MDSM$ is a centralized algorithm based on integer programming. It constructs an unconnected dominating set. The idea of using integer programming to find the minimum dominating set isn't new; in fact, there are several works that use different methods of integer programming to find both minimum dominating set and minimum connected dominating set [9]. However, all those past works, except from the centralized approach and the real-time constraints, consider only single layer graphs. Nevertheless, $FAST-MDSM$ comprises a departure from this literature.

III. PROBLEM FORMULATION

Let us start by briefly mentioning the concept of dominating set of graph [13]: a dominating set (DS) of a graph (i.e., the set of *dominators*) is a subset of the nodes of the graph such that the rest of the nodes are adjacent (i.e., into one hop distance) from some node(s) of the dominating set. There might exist more than one DS for a graph. In case the graph induced by the dominators is connected, then the DS is called connected DS (CDS). There might exist more than one CDS for a graph. For our purposes, we are interested in minimum CDS (MCDS), i.e., CDS with minimum cardinality. There might exist more than one MCDS for a graph. The problem of calculating a MCDS is NP-complete in the centralised setting [12].

In our IoBT context, we model an IoBT ad hoc network as a multilayer network, i.e., as a multilayer graph [7], [10], [19]. We assume the existence of only bidirectional links[3]. A multilayer 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) (i.e., with G_i nodes and E_i links), and a set of interlayer links $E^{ML} = \{E_{i,j} \subseteq G_i \times G_j; i,j \in \{1,\dots,n\}, i \neq j\}$. Figure 1 depicts such a network where for instance we can see a *layer* of soldiers, a layer of helicopters, the *intralayer* links connecting entities of the same layer, and *interlayer* links connecting entities belonging to different layers.

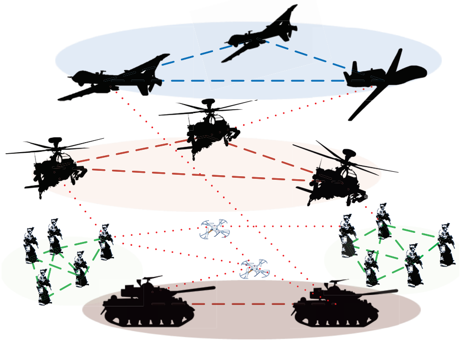

Fig. 1. A typical ad hoc network in the IoBT setting.

It is obvious that scalability issues [2] justify the requirement of finding a minimum cardinality CDS. Moreover, from the discussion in section I and in [2, SectionIII.B], where "...resilience and latency requirements for synthesizing a near-optimal network" [2] are emphasized, we conclude that our IoBT backbone must include as many nodes with many interlayer links as possible. This is because the existence of many nodes with "a lot" of interlayer links support low-latency communication among layers; think of them as the hubs encountered in complex networks that reduce the "degrees of separation". Also, if there are many nodes with "a lot" of interlayer links, then the danger of partition among layers is reduced significantly. Thus, we give the following definition (Definition 1) for the problem of calculating a MCDS for IoBT networks, and provide without proof its computational complexity (Proposition 1).

Definition 1 (Multilayer MCDS problem for IoBT): Solve the $MCDS$ for a multilayer 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 with many interlayer links is maximized, c) having only knowledge of the k-hop neighborhood around each node. Here, we set $k = 2$.

Proposition 1: The *Multilayer MCDS problem for IoBT* is NP-complete.

IV. PROPOSED ALGORITHMS

Due to the computational complexity of the problem, we strive to design efficient heuristic distributed algorithms to solve it; here we describe two such algorithms.

A. Distributed CDS in multilayer networks

In principle, any efficient heuristic algorithm for calculating a $MCDS$ seeks to detect in the topology strategically positioned nodes (connected to many other) in order to decrease the size of the obtained DS. In our case, we additionally seek such nodes with many interlayer links. So, we use the $clPCI$ *centrality* measure [19] to identify nodes with these two characteristics. Thus, we exploit the $clPCI$ measure and incorporate it into a distributed algorithm for computing a CDS. The algorithm will be called *Cross layer Connected Dominating Set* formation algorithm ($CCDS$).

In brief, the constituent main parts of $CCDS$ are the CDS construction and the redundant relay node pruning. Before these two steps take place, however, one more procedure evolves that is typical and common in (almost) all distributed algorithms for ad hoc networks with non GPS-enabled nodes. During this process, each node learns the topology of its neighborhood. For $CCDS$, each node learns the connectivity of all its neighbors up to its 2-hop neighborhood $N^2(u)$. Then, it calculates its own $clPCI$ index, it broadcasts its value to its neighbours and, by mutuality of the distributed protocol, it receives its neighbors' $clPCI$ values.

The CDS construction phase is based on a *source-initiated* relay node selection process that is executed by every node u, and is divided into two tasks, namely *neighbor prioritization* and *construction* task. In particular, each node u prioritizes its neighborhood in decreasing order of their $clPCI$ values and progressively selects from his $1-hop$ neighborhood $N(u)$ to include in its relay node set $R(u)$ the nodes with the largest $clPCI$ index value that cover at least one new node in the respective $2-hop$ neighborhood $N^2(u)$.

Then, a pruning phase follows, because the relay node selection process produces many redundant CDS nodes. To achieve a good balance between efficiency and overhead $CCDS$ makes use of the *restricted pruning Rule* k as this self pruning scheme, in general, is more efficient in reducing

---

[3] In principle, the paper ideas can be applied to multilayer networks with unidirectional links as well, but our algorithms need to be properly adjusted.

ISBN 978-3-903176-35-5 © IFIP

the relay node set than several existing schemes that ensure the broadcast coverage [28]. Notably, in the pruning rule we make use of connectivity as quantified by $clPCI$ as priority value in order to establish a total order among nodes that participate in the CDS. Connectivity has been proved to be the most efficient priority under all circumstances. The pseudo-code of $CCDS$ is presented in Algorithm 1.

Algorithm 1: CCDS

```
precondition : Known clPCI index values of nodes in
               $(N(u)) \wedge (N^2(u))$
postcondition: Completed MCDS election process
remarks      : mlNetwork G =(V, E) where $V$ and $E$ are
               vertex & edge set, $R(u)$ : relay node set of
               node $u \in V$, $M(u)$ : (T)rue/(F)alse indicator
               for node $u$ being a DS node.
repeat
    Add to $R(u)$ node $l \in N(u)$ which has the largest
     $clPCI$ and covers at least one new node in $N^2(u)$;
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 $S_{(u)}^{constrained} = u_1, u_2, \ldots, u_n \mid u_{k\ (1 \leq k \leq n)} \in$
     $N(u) \wedge N^2(u)$, $M(u_{k\ (1 \leq k \leq n)}) = T$,
     $clPCI(u) < clPCI(u_{k\ (1 \leq k \leq n)})$;
    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
        $M(u) = F$; Announce status change;
        Return;                        /* CDS Pruning */
    end
end
```

B. Centralized CDS in multilayer networks

$FAST-CMDSM$ is a centralized algorithm, i.e., it has a global view of the multilayer network topology at any time. Its constituent main parts are the MDS discovery, the CDS construction, and the redundant DS node pruning.

In order to calculate the minimum dominating set (MDS) it uses Integer Programming [16]. The CDS construction part concerns the addition in the DS of the least possible, *per node*, number of nodes in such a way that the 2-hop neighborhood of each respective node is covered and ultimately the multilayer network is connected, i.e., any two nodes can communicate through the DS nodes. The last part concerns the removal of any redundant DS nodes by discovering redundant paths within the netwok and thus substitute information flow through them.The pseudo-code of $FAST-CMDSM$ is presented in Algorithm 2.

C. Communication and computation complexities

In bidirectional networks, $CCDS$ requires 7 rounds to complete. If Δ is the maximum node degree in the network, the computation complexities of its constituent parts are $O(\Delta^2)$ for the $clPCI$ index calculation and $O(\Delta^3)$ for the relay node set election process and for the pruning phase.

Algorithm 2: FAST-CMDSM

```
precondition : All nodes are designated as dominators
postcondition: Completed MCDS election process
remarks      : mlNetwork G =(V, E) where $V$ and $E$ are
               vertex & edge set, $M(u)$ : (T)rue/(F)alse
               indicator for node $u$ to being a DS node, $Vm$ :
               DS node set, $MDS_V$ : Minimum DS node set
               of V, $CDS_V$ : Connected DS node set of V.
repeat
    if $\exists\ u_{j\ (1 \leq\ j\ \leq n)}\ \in V \mid$
    $d(u_j) = 1\ \&\ u_{i\ (1 \leq\ i\ \leq n,\ i\ \neq\ j)}\ \in\ N(u_j)$ then
        $M(u_{i\ (1 \leq i \leq n)})$ = T;
        if $u_{i\ (1 \leq i \leq n)} \notin Vm$ then
            Add node $u_{i\ (1 \leq i \leq n)}$ to $Vm$;
        end
    end
until all nodes at every layer have been examined
repeat
    if $M(u_{j\ (1 \leq j \leq n)})$ = $T$ then
        if $\exists\ u_{i\ (1 \leq i \leq n)} \in N(u_j) \& M(u_{i\ (1 \leq i \leq n)}) = T$ then
            $M(u_{j\ (1 \leq j \leq n)})$ = F; continue;
        else
            if $u_{j\ (1 \leq j \leq n)} \notin Vm$ then
                Add node $u_{j\ (1 \leq j \leq n)}$ to $Vm$;
            end
        end
    end
until no more nodes are added to $Vm$
$MDS_V$ = Minimize $\sum_{i=1}^{n} x_i$
Subject to $x_i + \sum_{j\ (u_j, u_i) \in E_k}^{n} x_j \geq 1\ \forall\ u_i \in V_{k\ (1 \leq k \leq n)}$
repeat
    Add to $CDS_V$ the least possible nodes from $N(u)$ that are
     needed to cover $N^2(u)$ neighborhood
until any node $u \in MDS_V$ has been examined
$CDS_V = Vm \bigcup (CDS_V - (CDS_V \bigcap Vm))$
Use Pruning in order to decrease the size of the $CDS_V$
```

The computation complexity of $FAST-CMDSM$ is exponential like all integer programming solvers that employ branch-and-cut algorithms.

V. PERFORMANCE EVALUATION

A. Competing algorithms

We compare the performance of the proposed algorithms, $CCDS$ and $FAST-CMDSM$. In order to highlight the utility of their pruning heuristics, we include in the comparison their annotated with an asterisk version of those, namely $CCDS^*$ and $FAST-CMDSM^*$ which concern their respective performance without the use of the pruning heuristic. Clearly for illustrative purposes, we present also the performance of $FAST-MDSM$, which constructs Minimum (unconnected) DS, in order to be used as a benchmark regarding the performance of the other competitors;i.e., we proved in [19] that any (unconnected) DS of size $\|DS\|$ can be turned into a CDS by adding $2 \times \|DS\|$ additional nodes in the DS in the worst case. TABLE I presents the unique characteristics of the competitors.

TABLE I
COMPETITOR CHARACTERISTICS

Competitor	**CDS Calculation**	**Pruning Heuristic**	**Complexities CPU**^{a} **/ Comm**^{b}
$CCDS$	Distributed	✓	Δ^3 / 7
$CCDS^*$	Distributed	–	Δ^3 / 7
$FAST-CMDSM$	Centralized	✓	exp / ^{c}
$FAST-CMDSM^*$	Centralized	–	exp / ^{c}
$FAST-MDSM$	Centralized	–	exp / ^{c}

[a] Δ is the maximum node degree in the network
[b] Number of messages transmitted per node to CDS conclusion
[c] Not applicable due to its centralized control

B. Simulation testbed

Due to the lack of available, real military networks, we developed a generator for multilayer networks in MATLAB, in order to create in an algorithmic way a variety of multilayer topologies. The generator should be able to generate topologies where the degree of a node, the diameter of each layer, the size of each layer, and the number of layers could vary after defining some parameters. The details of the generator can be found in [6], and here we present its basic features.

There are several wireless testbeds and several emulation environments for ad hoc network research [22]. The disadvantage of them is that they allow for experimentation with networks consisting of a few dozens of nodes. The requirement though of modern battlefields is to able to operate ad hoc networks consisting of twenty-fold more nodes; for instance a battalion would need a thousand nodes[4].

In our topologies each network layer consists of a set of nodes distributed in a two-dimensional plane. Each node has the same *maximum* transmission range, and by scaling, we set that all nodes have the same maximum transmission range equal to one. Every pair of nodes whose Euclidean distance is equal to or less than this maximum transmission range are assumed to be connected, i.e., they form a *Unit Disc Graph* (UDG). Moreover, in order to better approach reality where obstacles prohibit the direct communication between adjacent nodes, we used non-uniform intralayer models to distribute the nodes on the two-dimensional plane, as in [18]. 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), and the number of layers.

The interconnection of the different layers was done with the aid of two parameters: a) the number of links a node has towards nodes in different layers, b) the second parameter involves the distribution of interconnections towards the nodes within a specific layer. Given the above considerations, we apply the *Zipfian* distribution for our interconnectivity generator which can produce from uniform to highly skewed distributions for every parameter of interest. The desired skewness is managed by parameter $s \in (0, 1)$. We apply four distinct *Zipfian* distributions, one per parameter of interest:

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

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

1) Performance measures: The competing algorithms are compared in terms of the size of the CDS they generate. We say an algorithm is more efficient than another algorithm if it generates a smaller CDS [26], [30]. Additionally, an algorithm that manages to establish per node a relay set with larger minimum residual energy level is considered to be more energy efficient than another algorithm whose per node relay set includes relay nodes with less residual energy.

2) Datasets: We created datasets that simulate multilayer networks whose characteristics vary with respect to the topology density, the network diameter,the number of network layers and their size. The topology density impact is evaluated with 4-layer multilayer networks. Each layer is consisted of 50 nodes and its density varies, i.e., the mean degree of the participating nodes is 3, or 6, or 10, or 16, or 20. The network diameter impact is also evaluated with 4-layer multilayer networks. Each layer is consisted of 50 nodes and the mean degree of the participating nodes is 6. However, the diameter of each layer varies, i.e., it is 3, or 5, or 8, or 12, or 17. The impact of the number of layers is evaluated with multilayer networks consisting of 2, or, 3, or 4, or 5, or 7 layers. Each layer is consisted of 50 nodes and the mean degree of the participating nodes is 6. The impact of increasing the layer size is evaluated with 4-layer multilayer networks. The base layer is consisted of 50 nodes and each next layer is larger than the previous by 10%, or 20%, or 30%, or 50%, or 70%. The mean degree of the participating nodes in each layer is 6. Note that, the layers of a multilayer network share the same characteristics (density, diameter), unless otherwise specified.

---

[4] https://www.darpa.mil/news-events/2013-04-30.

Table II records all the independent parameters we used in our topology generator.

TABLE II
EXPERIMENTATION PARAMETERS VALUES

parameter	range	default
avg. node degree (D)	3, 6, 10, 16, 20	6
network diameter (H)	3, 5, 8, 12, 17	5
#network layers (L)	2, 3, 4, 5, 7	4
size of a layer relative to its adjacent layers	10%, 20%, 30%, 50%, 70%	-

C. Simulation results

We repeated each experiment 5 times, and recorded the variation in the performance, but each result was so tightly concentrated around the mean that the error bars are hardly recognizable in the plots.

1) Impact of topology density: Throughout this section, we consider the impact of topology density on the performance of each competitor. In Figure 2 we evaluate the *per layer* size of the CDS that each competitor creates. The overall observation is that the size of the CDS is almost a decreasing function with respect to the node density. That is due to the fact that the higher the network density the greater the coverage capability of the multilayer network nodes, and thus the smaller the size of the CDS. Interestingly, there is not a clear winner between $CCDS$ and $FAST-CMDSM$ as the topology becomes denser (degree ≥ 6) and both competitors present similar performance ($< 10\%$ variance). This is because in such topologies multiple redundant paths towards the nodes of the multilayer network exist and thus both pruning mechanisms work equally well. In sparse topologies (degree $= 3$), however, $FAST-CMDSM$ is up to 15% more efficient in terms of the CDS size compared to $CCDS$. This is due to the fact that in sparse connected networks, during the pruning process, the redundant paths are less and in order to be discovered it might be needed to go beyond the 2-hop neighborhood of a node. The centralized control of $FAST-CMDSM$ provides a clear advantage to him as its pruning mechanism has a broader overview of the network topology. When the pruning heuristic is not engaged we note that $CCDS^*$ and $FAST-CMDSM^*$ present similar performance (less than 10% variance) when degree ≤ 10 and the performance of the latter is up to 15% better to the former when degree> 10. However, these results are not interpretable as good because both algorithms do not perform well in terms of the, *per layer*, CDS size they construct; i.e., the, *per layer*, CDS size is up to 98% of that of the total number of nodes in that layer. However, that is common to happen in multilayer networks when traditional methods are used (2-hop neighborhood coverage) in order to assure connectivity within it. DS redundancy justifies the use of the pruning mechanism (up to 88% and 85% CDS size reduction for $CCDS$ and $FAST-CMDSM$, respectively, in this particular case).

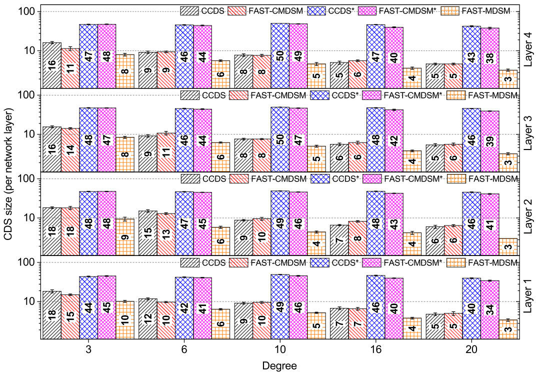

Fig. 2. Impact of topology density.

2) Impact of network diameter: In Figure 3 we evaluate the effect of the multilayer network diameter in the size of the CDS. The overall observation is that, as the network diameter increases the size of the constructed CDS for all competitors increases. The increment 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 a more demanding process, and hence more nodes are recruited. Focusing on the evaluation of the competitors, we observe that when we deal with bushy topologies (diameter ≤ 5) $CCDS$ presents up to 18% smaller CDS compared to $FAST-CMDSM$. This is due to the efficient pruning mechanism of $CCDS$ which is based on $clPCI$. In bushy topologies the removal from the CDS of a strategically located DS node may result to keep in the CDS a number of connected DS nodes, in order to maintain network connectivity. $CCDS$ quantifies the importance of each node in the multilayer network with the $clPCI$ and prioritizes the pruning process in such a way that the less important nodes (in terms of their $clPCI$ value) be removed first from the CDS. When diameter $= 8$ or diameter $= 12$ the competitors present similar performance (less than 10% variance). Notably, when dealing with long and skinny topologies (diameter $= 17$) $FAST-CMDSM$ outperforms $CCDS$ by 16% which is because of its centralized control which allows for better recognition - selection of the redundant paths inside the multilayer network. As for the respective pruning free version of the competitors both of them present unrealistic and unacceptable performance, i.e., almost all nodes are selected as DS nodes (the performance of the pruning mechanism for $CCDS$ and $FAST-CMDSM$ regarding the CDS size reduction is up to 83% and 79% , respectively, in this particular case).

3) Impact of number of layers: In this section, we consider the impact of the number of network layers on the performance of each competitor. Figure 4 presents the *per layer* size of the CDS that each competitor creates. First, we note that the *per layer* size of the CDS is irrespective to the number of layers. Counter-intuitive, we would expect the *per layer* CDS to decrease with respect to the number of layers because as

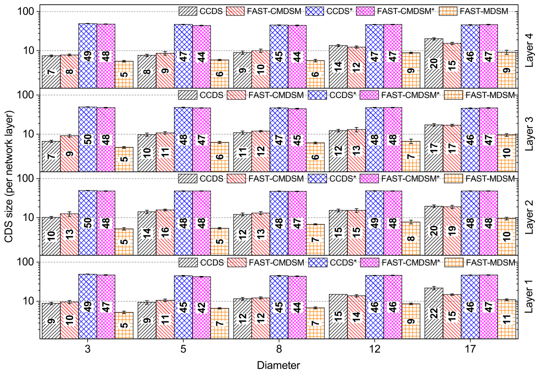

Fig. 3. Impact of network diameter.

the number of layers increases it increases also the number of interlinks among layers and the multilayer network becomes denser. Thus, we would expect the coverage capability of the nodes to increase accordingly. However, that does not happen and it is due to the multilayer network architecture. Focusing on the evaluation of the competitors, we observe that both $CCDS$ perform equally well (10% or less variance) when the number of layers is 5 or less. Notably, however, $CCDS$ is the champion algorithm compared to $FAST-CMDSM$, as it presents a better performance by 14%. Those results are consistent to the previous as we have already seen that both competitors perform well in dense topologies, which those under consideration are. The case where $CCDS$ outperforms $FAST-CMDSM$ is justified by its efficient pruning mechanism. As for the respective pruning free version of the competitors both of them continue their inefficient performance (the performance of the pruning mechanism for $CCDS$ and $FAST-CMDSM$ regarding the CDS size reduction is up to 79% and 75% , respectively, in this particular case).

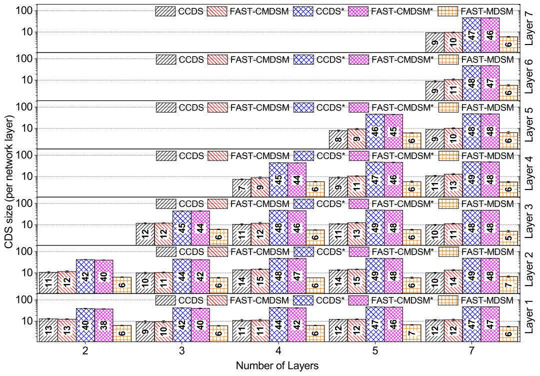

Fig. 4. Impact of network number of layers.

4) Impact of increasing the layer size: In this section, we consider the impact of increasing the layer size on the performance of each competitor. In Figure 5 we evaluate the *per layer* size of the CDS that each competitor creates. Note that the size of the CDS is an increasing function with respect to the increasing layer size. This happens because as the size of each layer increases it increases the need for more nodes to act as connectors and thus for more nodes for the CDS. Focusing on the evaluation of the competitors, we observe that when increasing the size of each subsequent layer by 30% or less then $CCDS$ outperforms $FAST-CMDSM$ from 11% $upto$ 16%. That is because in such dense topologies, the redundant paths remain inside the field of view of $CCDS$ (2-hop) and thus the pruning process is still efficient. On the other hand, when the increasing the size of each subsequent layer by 50% or more then $FAST-CMDSM$ outperforms $CCDS$ from 18% up to 21%. That is, because of the large difference in the cardinality of the multilayer network layers which equally results to a large number of interlayer links towards the other layers. In such a case in order to calculate the redundant paths and proceed with the pruning process a broader view of the network topology is needed, which justifies the better performance of the centralized controlled $FAST-CMDSM$. Finally, both the respective pruning free versions of the competing algorithms select almost all nodes as dominators and thus the results are not considered interpretable (the performance of the pruning mechanism for $CCDS$ and $FAST-CMDSM$ regarding the CDS size reduction is up to 80% and 83% , respectively, in this particular case).

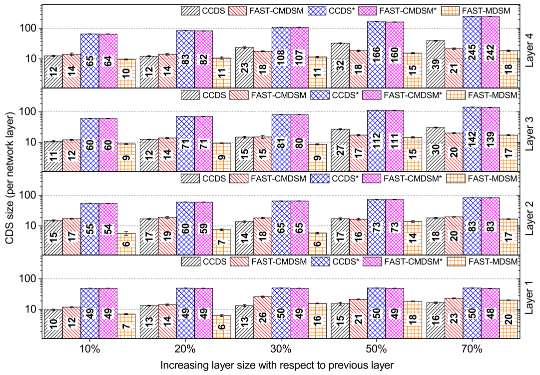

Fig. 5. Impact of increasing layer cardinality.

5) Energy adaptability of the competing algorithms: We repeated the majority of the previous experiments by taking into account the energy availability of each network node and we report the results in Figure 6; we see that $CCDS$ selects the most energy efficient CDS in, almost, any case, followed by $FAST-CMDSM$.

VI. CONCLUSION

The Internet of Battle(field) Things is a slowly emerging reality of the well known Internet of Things, but at a significantly larger scale, and with stringent requirements concerning robustness and latency. Its main goal is to carry out commander's intent in a safe, responsive and resilient manner.

In this article, we dealt with the problem of designing a small and resilient backbone for IoBT networks. We used the modelling strength of multilayer graphs to abstract the real topology of an IoBT network, and adopted the graph-theoretic concept of dominating sets to address our target. In

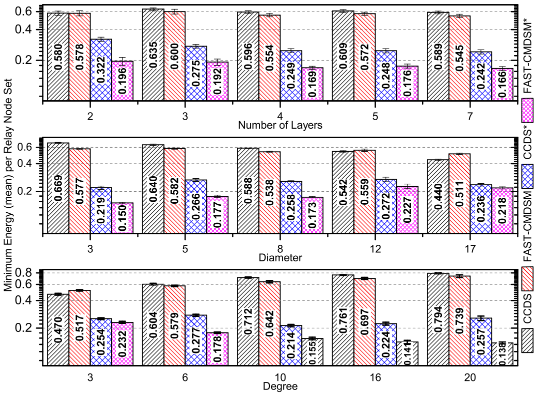

Fig. 6. Energy awareness of the competitors.

this context, we designed a distributed algorithm for calculating small-cardinality connected dominating sets for such multilayer networks where nodes located in the borders of and having connections to different/many layers were preferably selected as members of the dominating set, since they are able to support short communication latency, and significant tolerance to network partitioning attacks. We compared the developed algorithm to two state-of-the-art algorithms for computing connected dominating sets for multilayer networks using synthetically generated data across a range of topology characteristics. The proposed algorithm showed constantly better performance against its competitors.

Apart from future plans regarding algorithmic aspects and extensions of the proposed solution to unidirectional links, our future work includes the application of *tiny machine learning* in a federated fashion to other aspects of IoBT such as developing sophisticated distributed deep learning algorithms for heterogeneous learning nodes.

REFERENCES

[1] T. Abdelzaher and et al., "Toward and Internet of Battle Things: A resilience perspective," *IEEE Computer magazine*, vol. 51, no. 11, pp. 24–36, 2018.

[2] ——, "Will distributed computing revolutionize peace? The emergence of Battlefield IoT," in *Proceedings of the IEEE International Conference on Distributed Computing Systems*, 2018, pp. 1129–1138.

[3] N. Abuzainab and W. Saad, "Dynamic connectivity game for adversarial Internet of Battlefield Things Systems," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 378–390, 2018.

[4] ——, "A multiclass mean-field game for thwarting misinformation spread in the Internet of Battlefield Things," *IEEE Transactions on Communications*, vol. 66, no. 12, pp. 6643–6658, 2018.

[5] A. Azmoodeh, A. Dehghantanha, and K.-K. R. Choo, "Robust malware detection for Internet of (Battlefield) Things devices using deep eigenspace learning," *IEEE Transactions on Sustainable Computing*, vol. 4, no. 1, pp. 88–95, 2019.

[6] 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.

[7] 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. 1–222, 2014.

[8] A. Castiglione, K.-K. R. Choo, M. Nappi, and S. Ricciardi, "Context aware ubiquitous biometrics in edge of military things," *IEEE Cloud Computing magazine*, vol. 4, no. 6, pp. 16–20, 2017.

[9] N. Fan and J.-P. Watson, "Solving the connected dominating set problem and power dominating set problem by integer programming," in *Proceedings of the International Conference on Combinatorial Optimization and Applications (COCOA)*, 2012, pp. 371–383.

[10] M. J. Farooq and Q. Zhu, "On the secure and reconfigurable multilayer network design for critical information dissemination in the Internet of Battlefield Things (IoBT)," *IEEE Transactions on Wireless Communications*, vol. 17, no. 4, pp. 2618–2632, 2018.

[11] N. B. Gaikwad, H. Ugale, A. Keskar, and N. C. Shivaprakash, "The Internet-of-Battlefield-Things (IoBT)-based enemy localization using soldiers location and gunshot direction," *IEEE Internet of Things Journal*, vol. 7, no. 12, pp. 11 725–11 734, 2020.

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

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

[14] D. Katsaros, G. Stavropoulos, and D. Papakostas, "Which machine learning paradigm for fake news detection?" in *Proceedings of the IEEE/WIC/ACM Web Intelligence Conference (WI)*, 2019, pp. 383–387.

[15] A. Kott, A. Swami, and B. J. West, "The Internet of Battle Things," *IEEE Computer magazine*, pp. 70–75, 2016.

[16] J. C. Nacher, M. Ishitsuka, S. Miyazaki, and T. Akutsu, "Finding and analysing the minimum set of driver nodes required to control multilayer networks," *Nature Scientific Reports*, vol. 9, no. 576, 2019.

[17] I. Nasim and S. Kim, "Human EMf exposure in wearable networks for Internet of Battlefield Things," in *Proceedings of the IEEE Military Communications Conference (MILCOM)*, 2019.

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

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

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

[21] ——, "Energy-aware distributed edge domination of multilayer networks," in *Proceedings of the IEEE American Control Conference (ACC)*, 2019.

[22] K. N. Patel and R. H. Jhaveri, "A survey on emulation testbeds for mobile ad-hoc networks," *Procedia Computer Science*, vol. 45, pp. 581–591, 2015.

[23] S. V. Pemmaraju and I. A. Pirwani, "Energy conservation via domatic partitions," in *Proceeding of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC)*, 2006, pp. 143–154.

[24] S. Russel and T. Abdelzaher, "The Internet of Battlefield Things: The next generation of command, control, communications and intelligence (C3I) decision-making," in *Proceedings of the IEEE Conference on Military Communications*, 2018, pp. 737–742.

[25] I. Stojmenovic, M. Seddigh, and J. Zunic, "Dominating sets and neighbor elimination-based broadcasting algorithms in wireless networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 13, no. 1, pp. 14–25, 2002.

[26] Q. Tang, K. Yang, P. Li, J. Zhang, Y. Luo, and B. Xiong, "An energy efficient MCDS construction algorithm for wireless sensor networks," *EURASIP Journal on Wireless Communications and Networking*, no. 1, p. 83, 2012.

[27] D. K. Tosh, S. Shetty, P. Foytik, L. Njilla, and C. A. Kamhoua, "Blockchain-empowered secure Internet-of-Battlefield Things (IoBT) architecture," in *Proceedings of the IEEE Military Communications Conference (MILCOM)*, 2018.

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

[29] J. Wu and H. Li, "On calculating connected dominating set for efficient routing in ad hoc wireless networks," in *Proceedings of the Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIAL-M)*, 1999, pp. 7–14.

[30] 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.