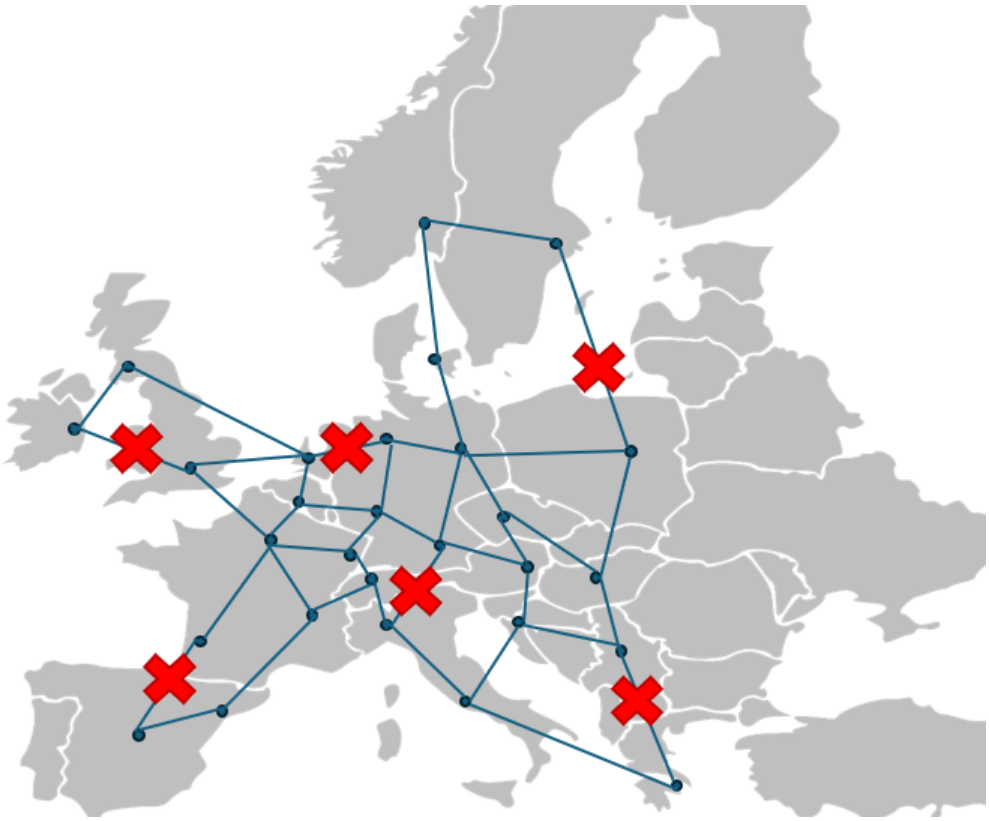
CHALMERS
UNIVERSITY OF TECHNOLOGY



Security Analysis of Distributed Consensus-based Network Architecture

Understanding and improving data center network resilience to targeted link cut attacks through mathematical modelling.

Master's thesis in Information and Communication Technology

EDGAR RODRIGO SÁNCHEZ COBOS

DEPARTMENT OF ELECTRICAL ENGINEERING

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2026
www.chalmers.se

Master's thesis 2026

Security Analysis of Distributed Consensus-based Network Architecture

Understanding and improving data center network resilience to targeted link cut attacks through mathematical modelling.

EDGAR RODRIGO SÁNCHEZ COBOS

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
Division of Optical Networks
Chalmers University of Technology
Gothenburg, Sweden 2026

Security Analysis of Distributed Consensus-based Network Architecture
Understanding and improving data center network resilience to targeted link cut attacks through mathematical modelling
EDGAR RODRIGO SÁNCHEZ COBOS

Supervisor: Ehsan Etezadi, Department of Electrical Engineering
Examiner: Marija Furdek Prekratic, Department of Electrical Engineering

Master's Thesis 2026
Department of Electrical Engineering
Division of Optical Networks
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Geo-representation of a graph the moment it is undergoing a link-cut attack.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2026

Security Analysis of Distributed Consensus-based Network Architecture
Understanding and improving data center network resilience to targeted link cut attacks through mathematical modelling
EDGAR RODRIGO SANCHEZ COBOS

Department of Electrical Engineering
Chalmers University of Technology

Abstract

Cloud computing is one of the most important communication paradigms supporting services related to data storage, execution of complex computations, hosting of applications, etc. The cloud is typically implemented over a network of data centers, which are giant, geographically remote warehouse-type facilities that host the computing and storage resources and run the related processes. The data centers are typically interconnected by optical fiber links capable of supporting high capacity and low latency communication. Their significance makes data center networks an alluring target for attacks aimed at disrupting services in order to cause damage or provoke chaos. The attacks are not constrained to the software realm, but can take place in the physical domain as well, e.g., by disabling data centers or cutting the optical fiber links to disrupt connectivity.
In distributed systems, consensus-based services are applied to take decisions regarding the distribution of resources, networking, or allocation of services. In consensus-based systems, instead of relying on a central system, there is a handful of selected nodes in the network that are part of an election system which takes decisions and enforces action via consensus. For a consensus to be reachable, a majority of the electing nodes need to be available and responding to requests initiated within the network. If a majority does not exist, the system will not work. Hence, the vulnerability lies in the capability of an attacking agent to disrupt network connectivity and/or the electing nodes such that a majority cannot be formed and consensus cannot be reached.
The main objective of this thesis is to provide an insight into the vulnerabilities of consensus-based systems to physical infrastructure attacks, i.e., fiber cuts and/or disabling of data centers. To this end, we develop an optimization model aimed at identifying the minimum effort needed by an attacker to prevent a majority from forming in a given data center network. The problem is formulated as an integer linear program and used to assess the vulnerability of three real-world reference network topologies of different sizes and characteristics, running consensus processes. Node placements guided by different topological principles, as well as random, are compared.
The developed framework and analysis of security vulnerabilities of distributed data center networks has the potential to support decisions related to the design of such networks and improve their robustness to network infrastructure attacks.

Keywords: ILP, network, security, distributed systems.

Acknowledgements

This piece of work is not only mine. It belongs to everybody who supported and continues supporting me in this journey I have taken into higher education.
To Dr. Marija Furdek at the Optical Networks group, for giving me the opportunity to take on this project; Dr. Ehsan Etezadi who so tirelessly helped me out through it all; and Dr. Mohit Chamania at Google in Germany, who served as a guide and advisor at all times.
To Chalmers University, Gothenburg, and Sweden, for receiving me with open arms and allowing me to blend into their community and to acquire not only valuable and everlasting academical experiences and knowledge, but societal and linguistic as well.
My mother and my father with their unconditional love and support since day one; my friends Luis, Germán, Victor, Alexis, Pato, Diego, and Yibriam; who have remained my awesome pals for so long despite an ocean distance in-between us.
To Lunita, Miguel, and Daniel, my fellow colleagues during university back home who still to this day carry on the spirit we held throughout our time as telecom students.
To my uncles, aunts, cousins, nieces, nephews, and grandmother, who I bear in my heart so dearly and serve as a force that pushes me forward in life. To the friends I have made here in Sweden, who made the life away from home worthwhile and have become a great influence in my life.
And last but not least, to my two beautiful dogs, Sisi and Cati, who I am sure are happy to see me make them proud from heaven.
Thank you all.

Edgar Rodrigo Sanchez Cobos, Gothenburg, June 2024

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

AAA Authentication, Authorization, & Audit

ACA Average Content Accessibility

ACL Access Control List

BCS Best Case Scenario

BC Betweeness Centrality

CAA-LAH Content-Accessibility-Aware Link Addition Heuristic

CAA-RAH Content-Accessibility-Aware Replica Addition Heuristic

CALA Consensus-Averting Link Attack

CANLA Consensus-Averting Node and Link Attack

CC Closeness Centrality

cDC core Data Center

CDN Content Delivery Network

DC Degree Centrality

DCN Distributed Control Node

eDC edge Data Center

HCC High Closeness Centrality

HDC High Degree Centrality

IDS Intrusion Detection Systems

ILP Integer Linear Programming

IoT Internet of Things

IPS Intrusion Protection Systems

LCC Low Closeness Centrality

LDC Low Degree Centrality

LP Linear Programming

MFA Multi-Factor Authentication

RCS Real Case Scenario

SLA Service Level Agreement

WCS Worst Case Scenario

μ-ACA Mean Content Accessibility

Nomenclature

Below is the nomenclature of indices, variables, and parameters that have been used throughout this thesis.

Indices

i,j	Indices for nodes within the network

Parameters

	Graph with a number of nodes V and a number of links E
N	Number of Distributed Control Nodes (DCNs) in the network
k	An auxiliary variable to define N as an odd value
c_{ij}	Cost associated to a single link cut
t_i	Cost associated to disabling a node
H_q	Occurrence of node disabling for the q iteration
R_q	Occurrence of link cuts in the q iteration
$\xi_{\delta u}$	Approximation to point of cost maximization for the hybrid attack

Variables

x_i, y_i, z_i	Binary variables representing the belonging of a node to a partition
δ_{ij}	Binary variable representing a link cut
u_i	Binary variable representing a node being disabled
q_i, a_i, w_i	Auxiliary variables related to u_i and x_i, y_i, and z_i, respectively

Contents

List of Figures

List of Tables

1

Introduction: Protecting Beyond the Software Realm

The world has entered a digital era where nearly every aspect of life relies on remote network connectivity, and where data is a valuable asset for businesses, governments, individuals, and other stakeholders. The networked society depends on the ability of various elements to connect to the internet and the cloud, where data is transmitted, processed, and stored. The cloud, in its physical form, consists of interconnected data centers: large facilities housing thousands of servers with vast data processing and storage capacities, and are physically connected using networking equipment and optical fibers that often span hundreds of kilometers.

Given the critical role of data centers in a world driven by political, economic, and social forces, these infrastructures become targets for various agents such as governments, hackers, hacktivists, and terrorist groups. These actors may seek to disrupt data center networks to achieve goals like interrupting the availability of services or illegitimately accessing data, thereby causing chaos that can damage the reputation, economy, vital services, or infrastructure of the targeted party. The potential economic and collateral damage caused by cyberattacks is enormous. For instance, the International Monetary Fund estimates that cyberattacks from 2017-2022 resulted in approximately $2.5 billion USD in economic damage across various industries, along with collateral effects such as the urgent need for consulting services and the reputational damage to the affected organizations [1].

To protect data center networks, security measures have evolved over time, following industry standards. These include the implementation of Access Control Lists (ACLs), Intrusion Detection Systemss (IDSs)/Intrusion Protection Systemss (IPSs), monitoring tools, and sandboxes, which are effective against software-based threats. However, a significant challenge remains: these infrastructures exist in the physical world. This introduces the following problem: ***Physical disruptions or attacks, such as link cuts or node tampering, can interrupt or degrade the communication and services provided by the infrastructure, regardless of the software security measures in place***.

This issue is of significant concern and has been the subject of research, such as in [2], which examines the vulnerabilities of Content Delivery Networks (CDNs).CDNs are low-latency, high-throughput structures used in applications like 4K video streaming, which rely on edge network infrastructure to reduce traffic running through the core network. The study explores the vulnerability of CDNs to targeted link-cut attacks and proposes a framework that counters such attacks without being detrimental to the user-experience of end-users and protecting the most critical links

in the infrastructure. The study relies on numerical simulations via an ILP model to find optimal solutions.This optimization technique helps identify exact solutions that offer insights into the inherent physical security of a network topology, which is crucial given the criticality and service level that this type of networks should be providing at all times.
As data center networks evolve, they increasingly decentralize certain applications to balance traffic more evenly across the infrastructure. An example of such application is the set of control processes that respond to infrastructure changes, such as node or link additions, disconnections, or changes in demand, which require redesigning and redistributing routing tables. This decentralization is achieved through the placement of DCNs, which collaboratively make decisions regarding network routing. A possible attack scenario targeting DCNs would be a targeted link-cut attack, with the goal to disrupt their cooperative decision-making by affecting their connectivity. In this context, and building upon previous research, this thesis aims at developing an ILP framework to assess the vulnerability of a DCN network to targeted link cut attacks. The framework is used to identify the optimal, i.e., cost-minimal cuts that disable consensus reaching in DCN networks. We also explore node disruption attacks that successfully partition the network, rendering DCN services infeasible. The thesis compares the effects of arbitrary DCN distribution to those with that incorporate awareness of attacks, analyzing how the different distributions respond to simulated attacks. The objective is to provide a deeper understanding of how strategic resource placement can enhance the physical security of distributed systems.
The following sections first provide an overview of the key concepts related to this project, followed by the formulation and design of the ILP model. Finally, an analysis of the results and conclusions drawn from this project are presented.

2

Background and Related Work

2.1 Distributed Data Center Architecture

Distributed data center architectures are important for handling the surge of new applications which demand high availability of content and low latency [3]. Applications such as video on demand, Internet of Things (IoT), remote computing, are examples of proliferating business models where a centralized cloud infrastructure cannot meet the expected performance [4]. When the cloud functionality is distributed across the infrastructure, a new concept known as "the fog" is established. It comprises nodes at the network edge, commonly referred to as "edge nodes", located close to or at the most external-facing realm and offering higher availability and lower latency.

In the case of DCNs, the distributed functionality is the management system [5]. An evenly distributed control system is able to detect and respond more rapidly to changes in the infrastructure, such as links being down, peaks in demand, application termination, or link addition.

In a distributed management system, nodes designated as DCNs need to be in constant communication with one another, exchanging constant update messages to reaffirm or refute the validity of tables which log their status [6]. When changes are perceived, the DCN which first detects an alteration of the network state recalculates new status tables and redistributes them to the other DCNs across the network. The other DCNs notify that these updates have been received and redistribute a message of confirmation to the origin DCN while they also calculate new tables. When all DCNs have calculated and redistributed their own tables and confirmed the reception of all other tables from the other DCNs, the recalculated tables are agreed upon via consensus, and if so, enforced.

This process relies on reachability and responsiveness of all DCNs. However, this is not always the case. A DCN can become unresponsive due to an overload of unrelated tasks or processes running over them, or undergoing maintenance. To account for this, the distributed management system only depends on the responsiveness of a majority of the DCNs in order for the proposed status changes to be enforced upon the active tables. Mathematically, this means that $\left\lfloor \frac{N}{2} \right\rfloor + 1$ DCNs need to be responsive to a recalculation request, where N is the total number of DCNs in the network. This consensus approach for a distributed control system is an implementation of the Paxos algorithm [7, 8], where a majority of $\{\frac{N}{2} + 1\} \in \mathbb{Z}^+$ is required to respond for the request to be considered, and the same majority must agree for the request to be approved. This thesis does not delve into the details of consensus-

based algorithms, assuming this basic explanation is enough to understand its initial premise, which is the usage and functioning of DCNs.

2.2 Data Center Security

The safeguarding and procurement of data center infrastructures is one of the top-prioritized and most budget-draining endeavors in the cloud service provisioning industry. The cloud has become an indispensable asset for multiple private and public sectors. Its protection is subject to security guidelines like the ones presented by cloud and data center security technology providers, such as VMWare [9], Checkpoint [10], Cisco [11], and Fortinet [12], focused on securing data center networks in the physical, virtual, and network domains.

2.2.1 Physical Security

Physical security refers to all the measures that should be taken in the physical domain when planning the deployment and operating data center architectures. Locations with a low risk of floods, earthquakes, or any other natural hazards should be chosen as a first step. Moreover, the considered locations need to be relatively secure from potentially disruptive human activity: they should be away from plane landing areas, chemical facilities, etc. Finally, security measures to guarantee a safeguarded perimeter for the premises need to be enforced: limited points of entry, physical barriers such as security checkpoints, fences, or physical locks, and other pertinent strategies should be put into place.

2.2.2 Virtual Security

Virtual security refers to all the software measures implemented in a data center to prevent attacks and defend against external agents who might try to disrupt the data center operations. Such measures can include, for example:

1. Monitoring systems that enable control over the people accessing the premises and their allowed actions through Authentication, Authorization, & Audit (AAA).
2. Up-keeping of patches, updates and upgrades of running software.
3. Enforcement of Multi-Factor Authentication (MFA) to guarantee the legitimacy of all changes or actions in the infrastructure.
4. The usage of sandboxing techniques to isolate the damage reach of intrusion attacks if it ever came to it.
5. The implementation of redundancy, to comply with the Service Level Agreements (SLAs) regarding downtime if a power outage or a system failure ever happens, by providing a backup of all system configuration, files, and data in a paired Data Center.

2.2.3 Network Security

Network security considerations encompass measures implemented at the network level, including, among other:

1. State-of-the-art firewalls, which can block undesired incoming traffic and keep track of network connections to verify the completeness of the information exchange transactions.
2. IPSs and IDSs for assessing potential misbehaviour and attacks.
3. Network segmentation, i.e., separation of network addresses per area, application or any other criterion, which allows for enclosure and detailed analysis of attacks while preserving the rest of the infrastructure.
4. Encryption methods to preserve integrity of exchanged information and prevent unauthorized parties from accessing it.

This thesis focuses on the interplay between the physical-layer security and the upper-layer network services. Our goal is to model the effect of link cuts and node disabling on the consensus reaching abilities in a data center network, and gauge the level of network resilience to different attacks. To this end, we model the problem of minimizing the cost of cutting links (or disabling nodes) that partitions a given network instance such that the consensus cannot be reached.

2.3 Integer Linear Programming (ILP)

The optimization problem of finding a minimal link cut that prevents a DCN network from reaching a consensus is modeled as an integer linear program. Linear Programming (LP) is an approach to finding optimal solutions to problems that can be modeled with a set of linear equalities and inequalities [13]. When formulating an LP model, key considerations include identifying the decision variables, defining the constraints that govern their relationships and permissible values, and specifying the objective function to be optimized. The objective function is evaluated subject to the constraints imposed on the decision variables and as a result a set of maximal and/or minimal values are found which satisfy all the aforementioned constraints. In an ILP, all variables and constants can only take integer values, hence the name *integer*. An illustrative example is provided to enhance understanding of this process:

1. Let us consider two decision variables, denoted as x and y. We would like to find a minimum value for the objective function $z = 5x + 4y$, subject to the following constraints:
   - $x + y >= 8$
   - $2x + y >= 10$
   - $x + 4y >= 11$
2. With these statements in place, the ILP model is ready to be used as input for a solver to find the optimal solution. The optimal value of the objective function in this minimization problem is **34**, for **x=2** and **y=6**.

When formulating an ILP model, the constraints should not be conflicting (e.g., $a >= b + c$, $b >= a$, and $c > 0$ are three incompatible constraints, making the optimization unfeasible). For a model feasibility check and solving, several linear

solvers and analytics tools are available (e.g. Gurobi [14], CPLEX [15], GLPK [16]), which integrate well with a vast number of the most popular programming languages (e.g. Java, Python, C++). The usage of an ILP model for the considered problem allows one to obtain exact optimal solutions for a real-world problem using a mathematical tool.

2.4 Related Work

The research that touches upon the analysis and proposal of remediation of targeted physical attacks varies in approach and the types of networks being analyzed. In [17], the threat analysis considers link-cut attacks against CDNs, defines a metric called Average Content Accessibility (ACA) to evaluate the network vulnerability to such attacks, and proposes replica placement approaches that maximize the ACA. In this work, different topologies are tested in three different scenarios: Best Case Scenario (BCS), Worst Case Scenario (WCS), and Real Case Scenario (RCS). BCS is defined as an upper-bound on the ACA when the topology is partitioned into components, while the WCS is defined as a lower-bound for the ACA. In RCS, the placement of the replica nodes is modeled via an exact placement based on metrics such as Betweeness Centrality (BC), Degree Centrality (DC), Closeness Centrality (CC) and clustering. The assessment is carried out for both simultaneous and sequential attacks. Using this model, the study shows that the placement of replicas based on the ACA metric improves content accessibility much more efficiently than when considering the usual structural metrics from the literature.
In [18], the ACA measure is further extended to Mean Content Accessibility (μ-ACA), which is the average content accessibility over an increasing link-cut attack. With this new measure that considers a set of link cuts whose size ranges from one to a given number, a heuristic algorithm called Content-Accessibility-Aware Link Addition Heuristic (CAA-LAH) ais proposed. It identifies a set of key links that, if added to the network, yield the highest improvement in the μ-ACA. The paper analyzes the impact of the different numbers of links added by CAA-LAH on the μ-ACA for different topologies and number of links cut, offering the possibility of an insightful analysis of different link addition strategies.
From this last research paper, a sturdy framework is proposed in [19] extending the use of the formulated μ-ACA, originally used in [18] for link addition by CAA-LAH, by proposing the Content-Accessibility-Aware Replica Addition Heuristic (CAA-RAH) strategy for CDN replica addition and placement. In this paper, an analysis of the effects on different topologies with different numbers of added links and replicas yields numerical results that demonstrate that both approaches highly increase the availability of content against these targeted attacks, offering a framework that can be leveraged to increase robustness of CDNs to physical-layer attacks.
In connection to this research papers detailing measures to determine the availability of content in network topologies under physical targeted attacks, in [2] explores the ideal placement of CDNs via an ILP model. The authors consider the cruciality of the nodes (which represents a higher impact in case of network fragmentation), and their hierarchy, distinguishing between the core Data Centers (cDCs), which store

all content and are reachable with a higher latency, and edge Data Centers (eDCs), which store only a fraction of all content with a lower latency. The ILP model for replica placement considers a budget for the placement of each type of node, and balances the robustness to link cuts with user-to-content distance. The set of identified solutions, characterized via their μ-ACA and user-to-content distance, is pareto-optimal, which means that they are the best solutions in spite of likely trade-offs. The simulation results show both improved network resilience and reduced load in the core network.
In this thesis, we leverage some of the key points proposed by these research papers to deliver an ILP model applied to a DCN network, and investigate the role of the number and the placement of DCN nodes in network resiliency to physical layer attacks.

3

Modelling and Optimization Framework

To assess the vulnerability of distributed data center networks to infrastructure attacks, we consider targeted link cuts and node disruption attacks aimed at disabling the consensus-reaching capabilities of the DCNs. We first consider a scenario with only link cut attacks. Node disruption attack capabilities are subsequently incorporated into the model and analyzed as a separate problem. An illustrative example is provided in each case, along with a performance assessment of the optimization approach.

3.1 The Consensus-Averting Link Attack (CALA) Problem

The problem considered in this thesis is modeled as the problem of identifying the min-cost CALA, defined formally as follows. The CALA problem takes the following inputs as given:

- The network topology is modeled as a graph $G(V, E)$, where V is a set of network nodes, interconnected by a set of links E implemented by bidirectional optical fibers. There are two types of nodes in the network: DCNs and non-DCNs. The DCNs are part of the management system and execute consensus-based processes, such as default routing changes or resource allocation; while also performing regular server functions. The non-DCNs perform only the regular node functions and do not participate in consensus reaching.
- The number of DCNs N, where N is odd, i.e., $N = 2k + 1$.

The problem is formulated as an ILP model with the objective of obtaining a minimal set of links whose cutting partitions the network such that the number of DCNs per partition is below the minimum required for reaching a consensus.
An illustrative example of the network topology considered in this work is depicted in Fig. 3.1. The DCN nodes are depicted with black color, while the non-DCN nodes are shown in grey.
To reach a decision through a consensus-based process, the majority of the decision-making entities, i.e., DCNs in the network must be available and responsive. Conversely, for an attack to prevent reaching a consensus, the network must be partitioned in a way that prevents the majority from forming. This translates into the following constraint: each partition formed by link cuts must contain at most $\lceil \frac{N}{2} \rceil - 1$ DCNs. When none of the network partitions contains more than $\lceil \frac{N}{2} \rceil - 1$

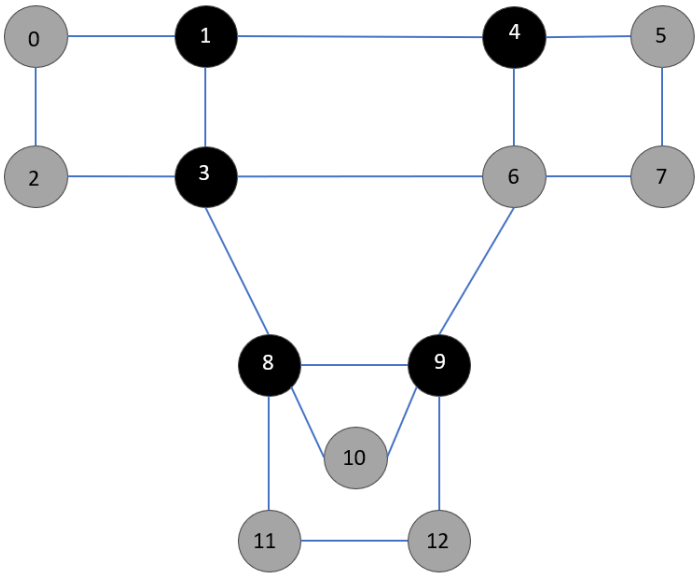


Figure 3.1: An illustrative example of a data center network topology with 13 nodes (5 of them DCNs) and 18 links.

DCNs, majority-based processes are disabled and a consensus cannot be reached. An attacker can use this model to their advantage and prevent a data center network from reaching consensus by cutting links and partitioning the topology such that the size of the largest partition does not exceed the $\lceil \frac{N}{2} \rceil - 1$ limit. To maximize the effectiveness of the attack, attacker's objective is to partition the network by cutting a minimum number of links.

Fig. 3.2 shows the illustrative example network partitioned in a way that prevents successful execution of consensus-based protocols by cutting four links (denoted by red lines) and creating three partitions (denoted with blue areas) whose sizes satisfy the $\lceil \frac{N}{2} \rceil - 1$ constraint. This solution represents the optimum, as it prevents consensus at a minimal number of cuts.

When assessing the vulnerability of consensus-based algorithms to physical-layer attacks and the attack efficiency, it is important to determine the minimum number of network partitions required to prevent reaching a consensus.Two partitions are typically insufficient, as one partition can likely retain a sufficient number of DCNs to achieve consensus. To assess this, let us consider a graph G in which all nodes are DCNs and participate in the consensus process, i.e., $|V| = N = 2k + 1$. Let us then consider that the optimal solution creates four partitions of any size and content of DCNs, denoted by (A, B, C, D). This solution can be reduced to a collapsed 4-node graph, with nodes denoted by(a, b, c, d), with an arbitrary degree of remaining connectivity within each sub-graph. Since there is no constraint which states that inside a sub-graph all nodes are connected to each other, any combination of the 4 collapsed nodes $[(a, b), (b, c), (c, d), (a, c), ...]$ is valid to consider for merging into a single bigger node and hence generating a three partition solution. The only way that a three-partition solution would not be sufficient to guarantee a complete disintegration of a DCN majority is if the size of the sub-partition created were >=k+1. In order to show that this >=k+1 sub-partition is not feasible, let us

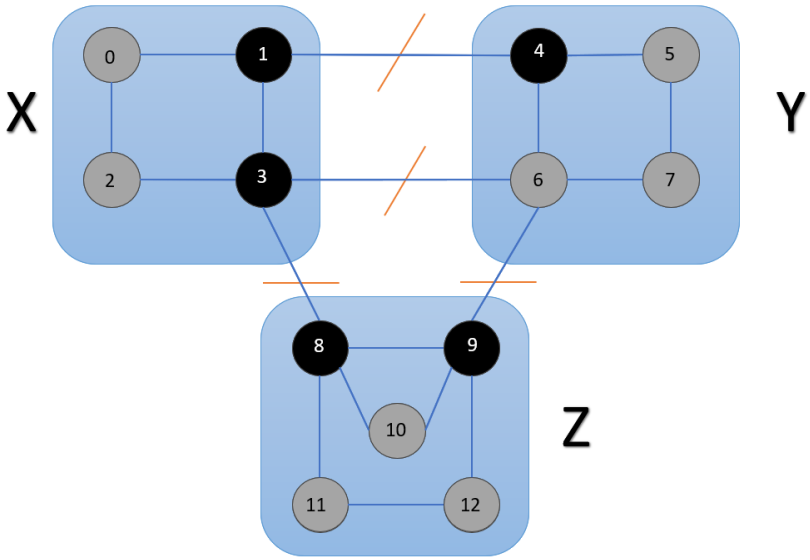


Figure 3.2: The illustrative example network divided into three partitions by a cost-minimal cut of 4 links that prevents forming of majority.

consider that:

$$a + b + c + d = 2k + 1 \tag{3.1}$$

To reduce the number of permutations for the proof, the 4 collapsed nodes representing the partitioned network are set to $a >= b >= c >= d$. With this consideration, a becomes the largest collapsed node, and therefore what needs to be demonstrated is that collapsing all the other smaller partitions result in sizes $> k + 1$.

$$b + c >= k + 1 \tag{3.2}$$

$$b + d >= k + 1 \tag{3.3}$$

$$c + d >= k + 1 \tag{3.4}$$

Through these three collapsed partition constraints, a solution for a is found:

$$2(b + c + d) >= 3k + 3 \tag{3.5}$$

Substituting from equation 3.1:

$$2(2k + 1 - a) >= 3k + 3 \tag{3.6}$$

Hence in the end:

$$-2a >= 1 - k; a <= \frac{k - 1}{2} \tag{3.7}$$

Given the invariant $a >= b >= c >= d$, with the solution for a:

$$a + b + c + d <= 4a <= 2k - 2 \neq 2k + 1 \tag{3.8}$$

With this it is now demonstrated that for every case 3 partitions are enough to make sure the $\lceil \frac{N}{2} \rceil - 1$ limit is respected.

3.2 ILP Formulation

The complete mathematical formulation of the CALA problem can be stated as follows, using similar notation to [5].

Input Parameters

- $G(V, E)$: bidirectional graph with a set of nodes V and a set of links E;
- N: the number of DCNs in the network, where $N = 2k + 1$;
- S: binary indicator of DCNs, where 0 denotes non-DCNs and 1 DCNs;
- c_{ij}: the cost of cutting a link between nodes i and j, set to 10.

Decision Variables

- x_i, y_i, $z_i \in \{1, 0\}$: the belonging of node $i \in V$ to one of the three partitions denoted as X, Y and Z. $x_i = 1$ if node i belongs to partition X, and 0 otherwise;
- $\delta_{ij} \in \{1, 0\}$: the link cuts. Equal to 1 if the link between nodes $i, j \in V$ is cut, and 0 otherwise.

Objective Function

$$\text{Minimize: } \sum_{e_{ij} \in E} c_{ij} \cdot \delta_{ij} \tag{3.9}$$

Subject to

$$x_i + y_i + z_i = 1 \tag{3.10}$$

$$\delta_{ij} \geq x_i - x_j \tag{3.11}$$
$$\delta_{ij} \geq x_j - x_i \tag{3.12}$$
$$\delta_{ij} \leq 2 - x_i - x_j \tag{3.13}$$
$$\delta_{ij} \geq y_i - y_j \tag{3.14}$$
$$\delta_{ij} \geq y_j - y_i \tag{3.15}$$
$$\delta_{ij} \leq 2 - y_i - y_j \tag{3.16}$$
$$\delta_{ij} \geq z_i - z_j \tag{3.17}$$
$$\delta_{ij} \geq z_j - z_i \tag{3.18}$$
$$\delta_{ij} \leq 2 - z_i - z_j \tag{3.19}$$

$$\forall v_i \in V, \sum S_i \cdot x_i \leq \lceil \frac{N}{2} \rceil - 1 \tag{3.20}$$

$$\forall v_i \in V, \sum S_i \cdot y_i \leq \lceil \frac{N}{2} \rceil - 1 \tag{3.21}$$

$$\forall v_i \in V, \sum S_i \cdot z_i \leq \lceil \frac{N}{2} \rceil - 1 \tag{3.22}$$

The objective of the model in 3.9 is to minimize the cost of link cuts that partition the network in a way that neither partition contains the majority of Distributed Control Nodes (DCNs). Constraint 3.10 guarantees that each node in the graph belongs to only one partition. Constraints 3.11 to 3.13 guarantee that link cuts can only happen between nodes belonging to different partitions: one node in the **X** partition and one in either **Y** or **Z**. If both x_i and x_j are equal to 1, δ_{ij} is equal to 0. Constraints 3.14 to 3.16 state the same for partition **Y**, and constraints 3.17 to 3.19 for **Z**. Finally, constraints 3.20 to 3.22 ensure that the number of DCNs per partition does not exceed $\lceil \frac{N}{2} \rceil - 1$.

3.3 Model Construction

The programming language chosen for this work was *Python*, given the strong familiarity with it and the already existing models that had been earlier developed at the department, allowing an easy support from the academics supervising this project. Two different solvers, i.e., CPLEX and Gurobi were assessed, and *Gurobi* was chosen due to seamless and efficient integration with Python, as well as the simple syntax.

Within Python, the module *Networkx* was imported, as it is a study and manipulation tool for network models. The topologies were built in GML (Graph Modeling Language) for easy syntax analysis and interpretation for Networkx. The build block of the model is depicted below. Access to the GitHub repository with the complete models and topologies is provided in Annex 1.

```
# Input topology as an argument

V=read(topology.gml)

#The model is initialized
mod=Model(name=mod)

#The fixed variables that will accompany the model are set

#total number of DCNs
N=5

#uniform cost associated to each link cut
c=10

#the decision variables are initialized and numbered
based in the number of nodes for x,y, z and delta_ij

x=mod.addVars(V.number_of_nodes(),name='x',vtype=GRB.BINARY)
y=mod.addVars(V.number_of_nodes(),name='y',vtype=GRB.BINARY)
z=mod.addVars(V.number_of_nodes(),name='z',vtype=GRB.BINARY)

delta_ij = mod.addVars(V.number_of_nodes(),
           V.number_of_nodes(),name='delta_{i}_{j}',
           vtype=GRB.BINARY)

#We set the objective function for the model to
optimize to and fix the type of optimization

obj_fun=sum(c*delta_ij[i,j] for i,j in V.edges())
nw_mod.setObjective(obj_fun, GRB.MINIMIZE)

#With this initialized we follow the next convention
for all constraints:
    - nw_mod.addConstr(math expression, "constraint name")

#Example using the constraint that associates only one partition to a node
```

```
for i in V.nodes():
    nw_mod.addConstr(x[i] + y[i] + z[i] == 1, 'constraint_one_subd{i}')

#After all the constraints have been added to the code,
the optimization of the model is initialized.

mod.optimize()
```

After the model is executed, an output with the objective function value and the variable values like in the example below is observed.

```
Optimization is done. Objective function value: 40.00
List of links cut:
                      0
delta_{i}_{j}[1,4] 1.0
delta_{i}_{j}[3,6] 1.0
delta_{i}_{j}[3,8] 1.0
delta_{i}_{j}[4,1] 1.0
delta_{i}_{j}[6,3] 1.0
delta_{i}_{j}[6,9] 1.0
delta_{i}_{j}[8,3] 1.0
delta_{i}_{j}[9,6] 1.0
Nodes in partition X:
[0, 1, 2, 3]
Nodes in partition Y:
[8, 9, 10, 11, 12]
Nodes in partition Z:
[4, 5, 6, 7]
```

Retrieving the visualization of the graph used as an illustrative example in Fig. 3.1, where DCNs are nodes 1, 3, 4, 8, and 9, the solution from the output above is depicted in Fig. 3.2. The network is partitioned into three partitions by four link cuts: link 1-4, 3-6, 3-8, and 6-9. The cost associated to cutting a link is 10, which means that the objective function value obtained for the optimal solution to the CALA problem for this graph is 40.

3.4 Performance Assessment

The framework developed in this project is applicable to any given network topology that follows the same definition principles defined before. To validate the applicability of the model, performance assessment was conducted using topologies that accurately represent real-world scenarios. To achieve that, the openly available SNDLib [20] from Poznan University of Technology, which holds data sources for communication network optimization, was utilized.
Three topologies representing existing networks were selected: one in the European Union (EU) and two in North America (Atlanta; US & Canada). These topologies vary in size to reflect diverse network conditions. The graphs are illustrated along with the corresponding solutions of the ILP model. In this analysis, the DCN selection is performed via an arbitrary method: given $N \in \mathbb{Z}^+$, the DCNs are selected randomly among the network nodes. For the sake of simplifying the analysis in this section, we set $N = 5$ for all network topologies.

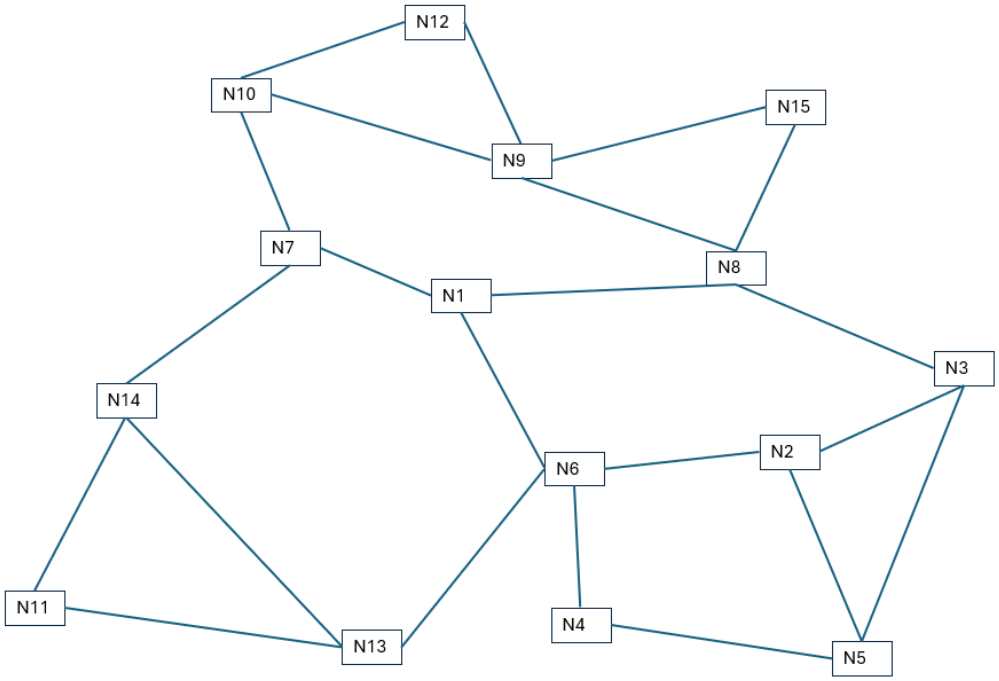


Figure 3.3: Network topology of the city of Atlanta with 15 nodes and 22 links.

3.4.1 Atlanta Graph

As a first example, we use the graph representing the network of the city of Atlanta with |V|=15 and |E|=22, shown in Fig. 3.3. An example output obtained after solving the ILP model for this graph with arbitrary DCN distribution is provided below. The default values of $N = 5$ and $c_{ij} = 10$ are used.
In the code, the node names (e.g., N11) are replaced with indices spanning from 0 to |V|-1. These indices are used in the model and directly related to the variables which depend on them ($\delta_{ij}, x_i, y_i, z_i$).
Fig. 3.4 shows the same topology using node indices instead of tags, as well as the DCNs selected arbitrarily for this example run depicted in white: (4) N5, (7) N8, (8) N9, (10) N11, and (14) N15.

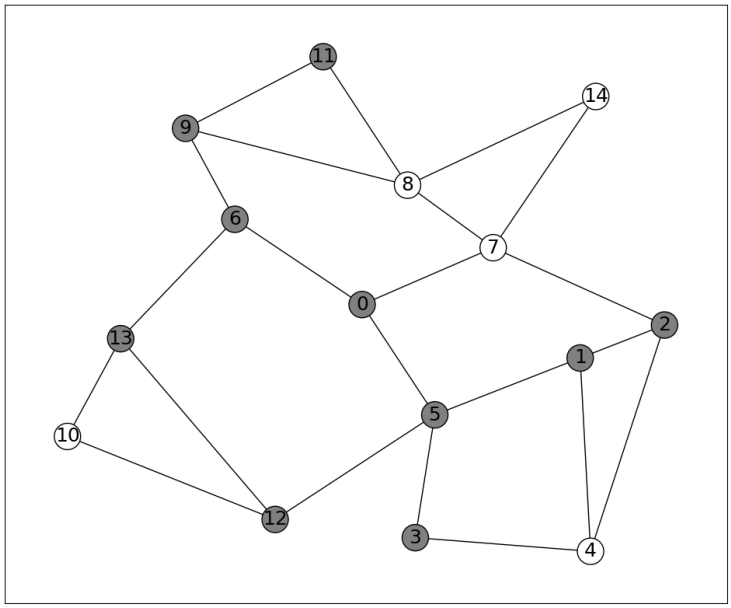


Figure 3.4: The Atlanta graph with indices instead of tags and arbitrarily selected five DCNs.

The result obtained by solving the ILP for this problem instance is reported as follows.

```
Optimization is done. Objective function value: 50.00
List of links cut:
                        0
delta_{i}_{j}[6,9] 1.0
delta_{i}_{j}[7,8] 1.0
delta_{i}_{j}[7,14] 1.0
delta_{i}_{j}[8,7] 1.0
delta_{i}_{j}[9,6] 1.0
delta_{i}_{j}[10,12] 1.0
delta_{i}_{j}[10,13] 1.0
delta_{i}_{j}[12,10] 1.0
delta_{i}_{j}[13,10] 1.0
delta_{i}_{j}[14,7] 1.0
Nodes in partition x:
[0, 1, 2, 3, 4, 5, 6, 7, 12, 13]

DCN nodes in this partition
4 N5
7 N8

Nodes in partition y:
[10]
DCN nodes in this partition
10 N11
Nodes in partition z:
[8, 9, 11, 14]
DCN nodes in this partition
8 N9
14 N15
```

A visualization of the result with the three created partitions is shown in Fig. 3.5.

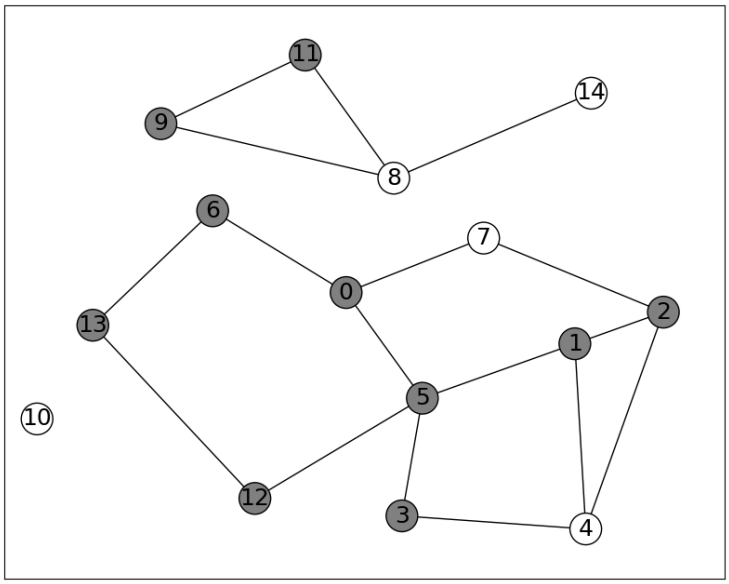


Figure 3.5: An example result for the link-cut attack against the Atlanta network.

Given that this experiment utilizes an arbitrary DCN placement scheme, there are numerous possible combinations for selecting the DCNs. For example, for N=5, there are 3003 possible DCN placement combinations for this graph. Hence, the example value obtained for the single instance shown above is not enough to acquire an encompassing view of the network vulnerability to link cut attacks. A statistical analysis considering all possible placements of 5 DCNs is needed to determine the performance for different DCN placements, as shown in Fig. 3.6.

The data provided by Fig. 3.6 shows that there are only two values for the considered graph: 40 (i.e., 4 link cuts) and 50 (i.e., 5 link cuts). With the value of 40 obtained in 1644 instances and the value of 50 obtained in the remaining 1359, there is a 9.4905% difference between the two cost values, which translates into the following: if an arbitrary placement of DCNs were chosen for this particular graph, the likelihood of an attack having an optimal cost equal to the lower bound value of 40 is 54.745%. A minimum-cost attack comprises either 4 or 5 link cuts out of the 22 links in E, which translates into the attacker obtaining access to only $\sim$18% or $\sim$23% of links, respectively. The obtained values and observations are compared with those from non-arbitrary DCN placement executions and varying numbers of DCNs in the results section.

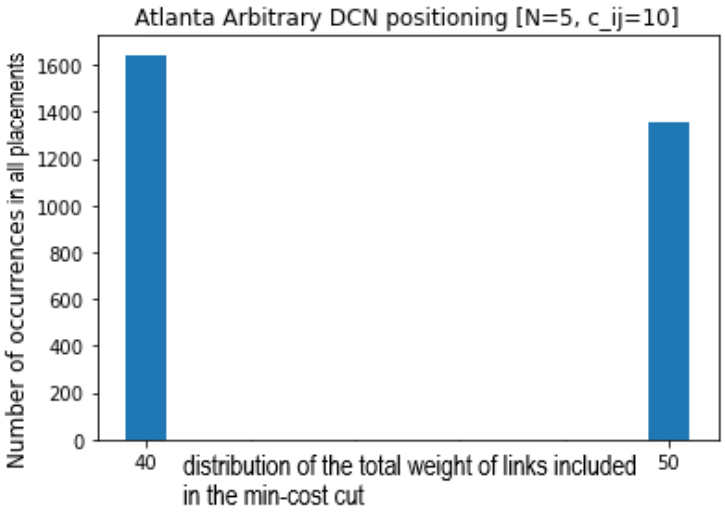


Figure 3.6: The distribution of minimal attack cost for all possible DCN placements for the Atlanta graph and N=5.

3.4.2 EU Graph

As a second example, we use the graph representing the network of the EU with |V|=28 and |E|=41, shown in Fig. 3.7.

Figure 3.7: EU network topology with 28 nodes and 41 links.

An example output obtained after solving the ILP model for this graph with arbitrary DCN placement is provided below. The default values of $N = 5$ and $c_{ij} = 10$ are used. In the code, the node names (e.g., Stockholm) are replaced with indices spanning from 0 to |V|-1. These indices are used in the model and directly related to the variables which depend on them ($\delta_{ij}, x_i, y_i, z_i$). Fig. 3.8 shows the same topology

using node indices instead of tags, as well as the DCNs selected arbitrarily for this example run depicted in white: (4) Berlin, (9) Dublin, (12) Hamburg, (13) London, and (24) Vienna.

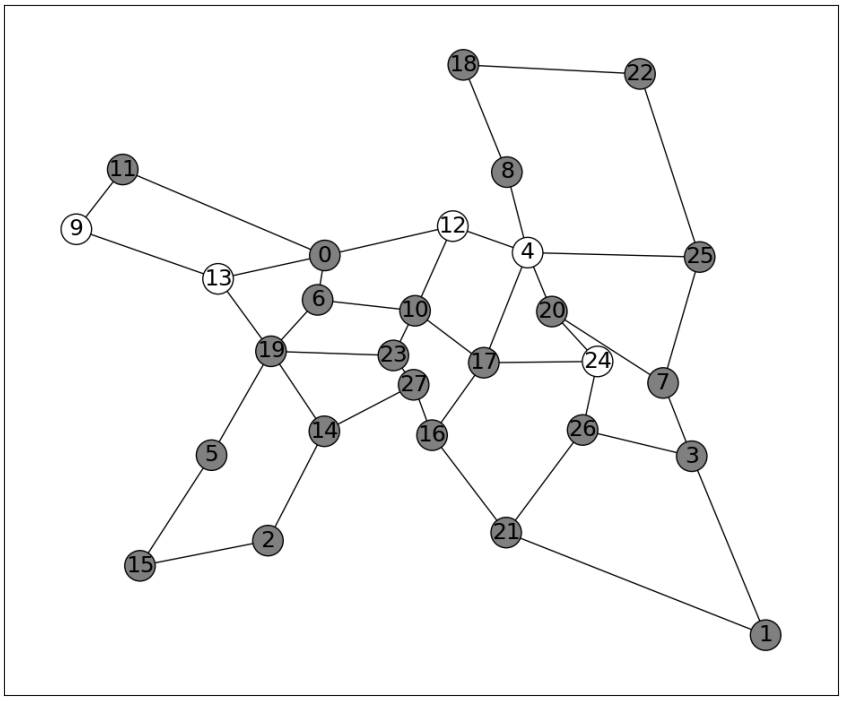


Figure 3.8: The EU graph with node indices instead of city names and arbitrarily selected five DCNs.

The result obtained by solving the ILP for this problem instance is reported as follows.

```
Optimization is done. Objective function value: 50.00
List of links cut:
                                    0
delta_{i}_{j}[0,6] 1.0
delta_{i}_{j}[0,12] 1.0
delta_{i}_{j}[4,12] 1.0
delta_{i}_{j}[6,0] 1.0
delta_{i}_{j}[10,12] 1.0
delta_{i}_{j}[12,0] 1.0
delta_{i}_{j}[12,4] 1.0
delta_{i}_{j}[12,10] 1.0
delta_{i}_{j}[13,19] 1.0
delta_{i}_{j}[19,13] 1.0
Nodes in partition X:
[0, 9, 11, 13]
DCN nodes in this partition
9 Dublin
13 London
Nodes in partition Y:
[12]
DCN nodes in this partition
12 Hamburg
```

```
Nodes in partition Z:
[1, 2, 3, 4, 5, 6, 7, 8, 10, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26,
    ↪ 27]
DCN nodes in this partition
4 Berlin
24 Vienna
```

A visualization of the result with the three created partitions is shown in Fig. 3.9.

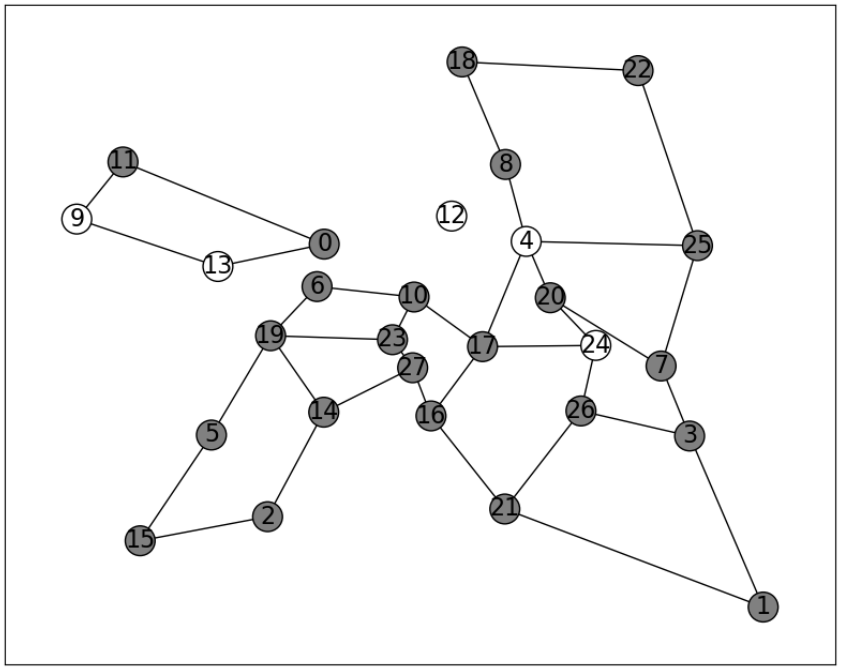


Figure 3.9: The min-cost link cut attack against the EU network example.

Given that this experiment utilizes an arbitrary DCN placement scheme, there are numerous possible combinations for selecting the DCNs. For example, for N=5, there are 98280 possible DCN placement combinations for this graph. Hence, the example value obtained for the single instance shown above is not enough to obtain an encompassing view of the network vulnerability to link cut attacks. A statistical analysis considering all possible placements of 5 DCNs is needed to determine the performance for different DCN placements, as shown in Fig. 3.10.
The data provided by Fig. 3.10 show four values for the considered graph: 30 (i.e., 3 link cuts), 40 (i.e., 4 link cuts), 50 (i.e., 5 link cuts), and 60 (i.e., 6 link cuts). The value of 30 is obtained in 600 instances, the value of 40 in 10344, the value of 50 in 63534, and the value of 60 in the remaining 23802. The most recurring value (50) has a percentage of occurrence of 64.65%, while the lowest possible minimum (30) occurs in only 0.61% of the cases for this graph. The 4 identified minima translate to access to ∼7% of the links for the 3-link cut solution, ∼10% of the links for the 4-link cut, ∼12% of the links for the 5-link cut, and ∼15% of the links for the 6-link cut attack. The six-link cut solution corresponding to the associated cost value of 60 is the best one from the network operator point of view, as it requires the maximum effort from an attacker to prevent consensus. The obtained values and observations are compared with those from non-arbitrary DCN placement executions and varying

numbers of DCNs in the results section.

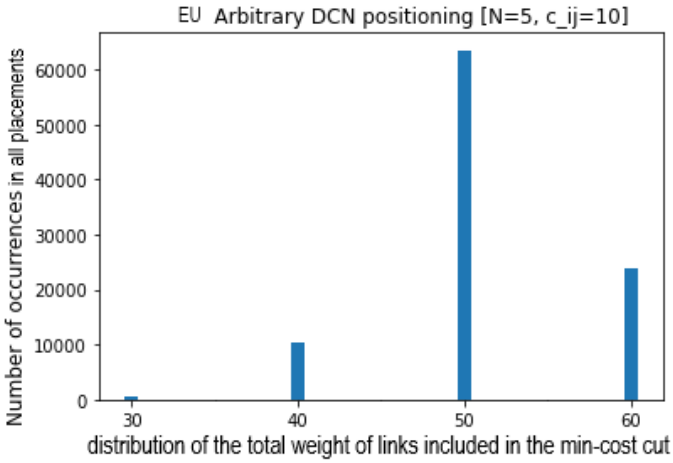


Figure 3.10: The distribution of minimal attack cost for all possible DCN placements for the EU graph and N=5.

3.4.3 US-Canada Graph

As a final performance assessment we use the graph representing the network of the city of Atlanta with |V|=39 and |E|=61, shown in Fig. 3.11.

Figure 3.11: US-Canada network topology with 39 nodes and 61 links.

An example output obtained after solving the ILP model for this graph with arbitrary DCN distribution is provided below. The default values of $N = 5$ and $c_{ij} = 10$ are used.
In the code, the node names (e.g., Toronto) are replaced with indices spanning from 0 to |V|-1. Fig. 3.12 shows the same topology using node indices instead of tags,

as well as the DCNs selected arbitrarily for this example run depicted in white: (0) Vancouver, (15) StLouis, (17) Cleveland, (19) Montreal, and (27) Toronto.

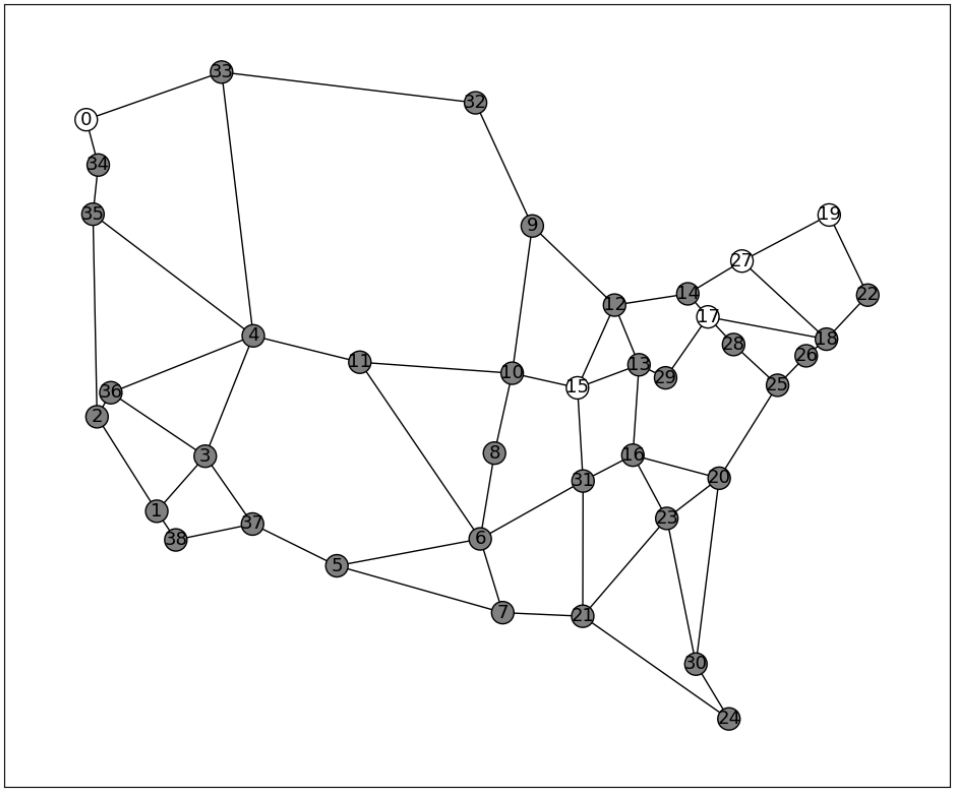


Figure 3.12: The US-CA graph with indices instead of tags and arbitrarily selected five DCNs.

The result obtained by solving the ILP for this problem instance is reported as follows.

```
Optimization is done. Objective function value: 50.00
List of links cut:
                        0
delta_{i}_{j}[0,33] 1.0
delta_{i}_{j}[14,27] 1.0
delta_{i}_{j}[18,27] 1.0
delta_{i}_{j}[19,22] 1.0
delta_{i}_{j}[22,19] 1.0
delta_{i}_{j}[27,14] 1.0
delta_{i}_{j}[27,18] 1.0
delta_{i}_{j}[33,0] 1.0
delta_{i}_{j}[34,35] 1.0
delta_{i}_{j}[35,34] 1.0
Nodes in partition x:
[0, 34]
DCN nodes in this partition
0 Vancouver
Nodes in partition y:
[19, 27]
DCN nodes in this partition
19 Montreal
```

```
27 Toronto
Nodes in partition z:
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 20, 21, 22, 23,
    ↪ 24, 25, 26, 28, 29, 30, 31, 32, 33, 35, 36, 37, 38]
DCN nodes in this partition
15 StLouis
17 Cleveland
```

A visualization of the result with the three created partitions is shown in Fig. 3.13.

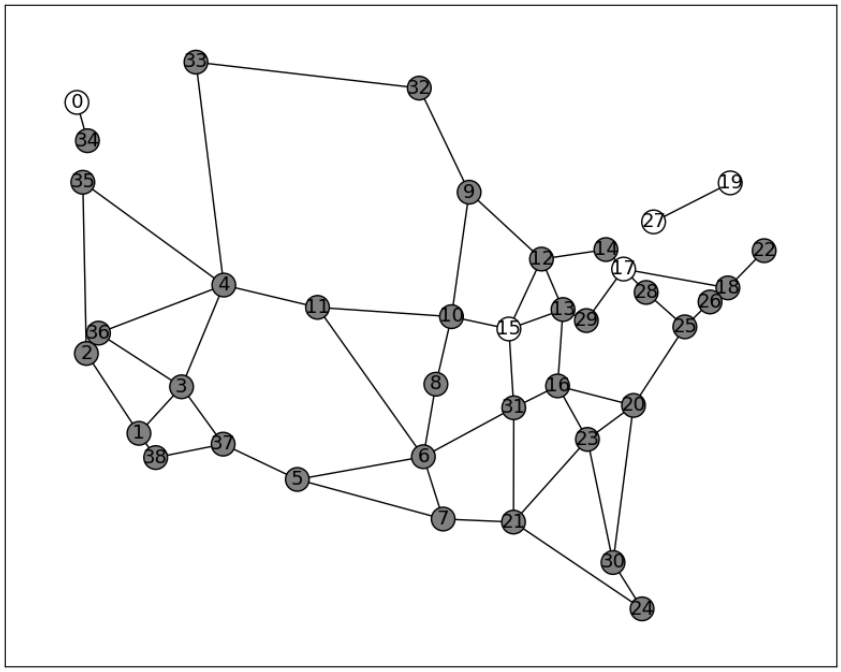


Figure 3.13: An example result for the link-cut attack against the US-Canada network.

As in the previous example, we utilize an arbitrary DCN placement scheme with numerous possible combinations for selecting the DCNs. For example, for N=5, there are 575757 possible DCN placement combinations for this graph. Hence, the example value obtained for the single instance shown above is not enough to obtain an encompassing view of the network vulnerability to link cut attacks. A statistical analysis considering all possible placements of 5 DCNs is needed to determine the performance for different DCN placements, as shown in Fig. 3.14.

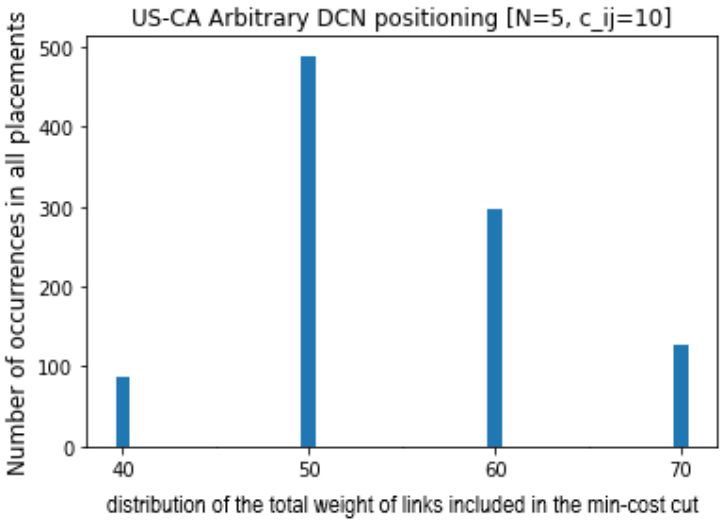


Figure 3.14: The distribution of total attack budget over all possible DCN placements for the US-CA graph and N=5.

Fig. 3.14 shows a trend to an occurrence of only four values for the current graph: 40 [4 link cuts], with an occurrence of 49854 times, 50 [5 link cuts] with an occurrence of 286900, 60 [6 link cuts] with an occurrence of 177800, and 70 [7 link cuts] with an occurrence of 61203. The most recurring value (50) has a percentage of occurrence of 49.83%, with the lowest possible achievable minimum occurring only a 8.65% of the times. The 4 identified minima translate to access to ∼7% of the available links for the 4 cut solution, ∼8% of the available links for the 5 cut one, ∼10% of the available links for the 6 cut one, and ∼11% of the available links for the 7 cut one. In this graph, the 7-cut solution with a total associated weight of 70 is the best-performing set of DCN distribution as it requires the most effort to prevent consensus from forming among all possible $N = 5$ options. These objective function values and related observations are compared against non-arbitrary DCN placement executions in the results section.

3.5 The Consensus-Averting Node and Link Attack (CANLA) Problem

In addition to cutting the links, an attacker may have knowledge of the DCN locations and use it when trying to disable the consensus-based functionalities by, e.g., cutting the power supply to a node, reducing its functionalities to non-management-related functions running on reserve power. This reduction in the number of active DCNs may consequently reduce the effort associated with cutting links.

These considerations give rise to a new version of the problem, which we refer to as the Consensus-Averting Node and Link Attack (CANLA) problem. The inputs to the problem are the same as for CALA, with added flexibility of disabling nodes as well as links. The objective is again to minimize the total cost of an attack that prevents the network nodes from reaching consensus. To this end, we model two types of cost: c_{ij} is the cost of cutting a link between nodes i and j, and t_i represents the cost of disabling node i. Modeling both costs separately and investigating the effects of the relation between them helps us understand the attack decisions better. We investigate three different cost relation scenarios ($t_i = c_{ij}$, $t_i > c_{ij}$, and $t_i >> c_{ij}$), which allow us to explore how an attacker's optimal mix of actions shifts as node-disabling gradually becomes more expensive than link-cutting. This analysis is essential to understand the structural weaknesses an attacker will exploit under different cost relation constraints.

As mentioned in the introduction, victims of cyberattacks suffer various types of damage, including financial losses which can be quantified [1]. However, there are no known economic models of costs of performing various physical attacks, like targeted link-cut attacks or node service disruption. The cost of an attack depends on several factors such as location of the nodes, terrain in which the links are placed, the knowledge or investment in knowledge the attacker might have, the tools they might have access to, the access viability to the locations of the links or DCNs, and many other. Despite of the lack of attack cost models, we assume that disabling of nodes would be at least as costly as cutting a link, and potentially much costlier.

In our analysis, we introduce parameter p, which denotes the maximum allowable DCNs in any partition after an attack. We revisit the constraint introduced in section 3.1 as well as the illustrative example from Fig. 3.1 with $N = 5$ and $c_{ij} = 10$. A maximum value of $p = 2$ DCNs is allowed to be placed into one single partition when $N = 5$ to satisfy the conditions on consensus prevention. The optimal solution of the CALA problem, illustrated in Fig. 3.2, comprised 4 link cuts with the total cost of 40. Considering the scenario where $c_{ij} = t_i = 10$, the same effect can be achieved by disabling 3 nodes, with a total cost of 30. This solution is illustrated in Fig. 3.15 where the disabled nodes are shown with white color, the active DCNs with black, and the non-DCN nodes with gray.

To model the scenario where $t_i > c_{ij}$, we consider $t_i = 15$ and $c_{ij} = 10$, which relates to a 50% higher cost of disabling a node than disabling a link. For the $t_i >> c_{ij}$ scenario, we consider $t_i = 30$ and $c_{ij} = 10$, corresponding to a 200% difference. These values are used throughout the remainder of this work.

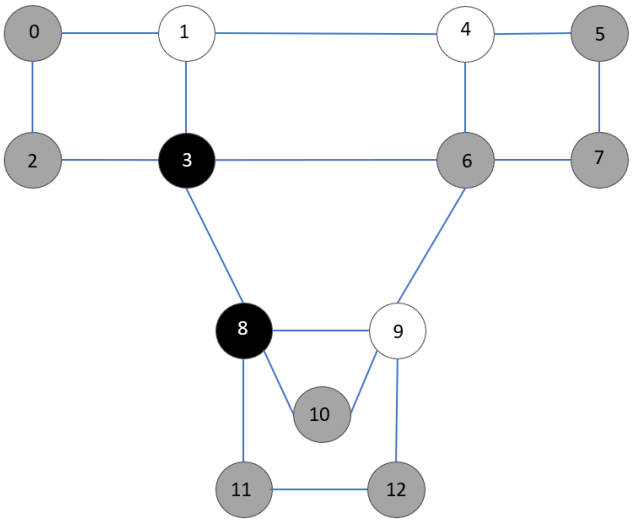


Figure 3.15: The illustrative example network divided into one partition with a three node disabling, cost-minimal, hybrid approach when $t_i = c_{ij}$.

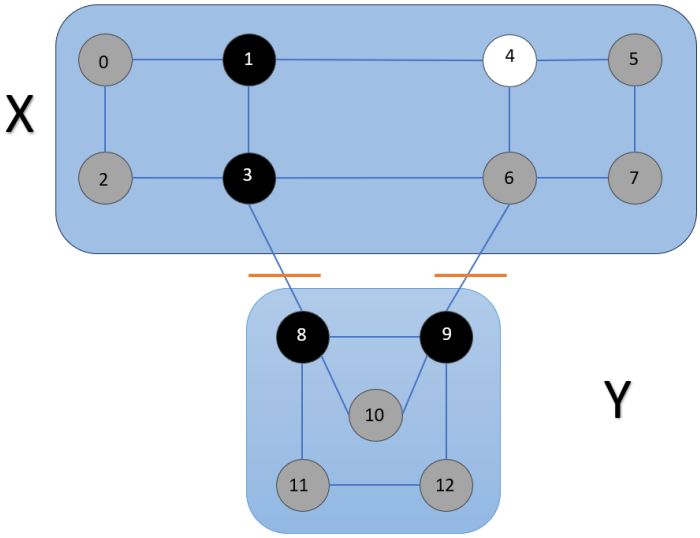


Figure 3.16: The illustrative example network divided into two partitions with a single node disabling, two-link cut, cost-minimal, hybrid approach when $t_i > c_{ij}$.

Fig. 3.16 shows the optimal solution in the $t_i > c_{ij}$ case, where a hybrid solution leveraging both types of physical attacks offers a two partition solution which still guarantees the constraint for $\leq \lceil \frac{N}{2} \rceil - 1$ active DCNs per partition (≤ 2).

3.6 ILP Formulation

When solving the CANLA problem, the objective is to minimize the total cost incurred by cutting the links and/or disabling the DCNs that prevents forming majority and reaching consensus. The ILP formulation uses the same variables as the CALA model, and introduces one additional input parameter and four variables.

Additional Input Parameters

- t_i: cost unit related to the disabling of a DCN, which will be tuned to three cases for comparative analysis: $t_i = c_{ij}$, $t_i > c_{ij}$, and $t_i >> c_{ij}$.

Variables

- $u_i \in \{1, 0\}$: the DCN status: equal to 1 if DCN $i \in V$ is disabled, and 0 if it is active.
- q_i, a_i, $w_i \in \{1, 0\}$: auxiliary variables related to u_i, which guarantee that u_i can only be equal to 1 if its related partition variable (x_i,y_i,z_i) is equal to 1.

Objective Function

$$\text{Minimize: } \sum_{e_{ij} \in E} c_{ij} \cdot \delta_{ij} + \sum_{v_i \in V} t_i \cdot u_i \tag{3.23}$$

Subject to

Constraints 3.10-3.19 and 3.24 to 3.32.

$$2q_i \leq u_i + x_i \quad (3.24) \qquad 2a_i \leq u_i + y_i \quad (3.26) \qquad 2w_i \leq u_i + z_i \quad (3.28)$$

$$1 + 2q_i \geq u_i + x_i \quad (3.25) \qquad 1 + 2a_i \geq u_i + y_i \quad (3.27) \qquad 1 + 2w_i \geq u_i + z_i \quad (3.29)$$

$$\forall v_i \in V, \sum S_i \cdot (x_i - q_i) \leq \lceil \frac{N}{2} \rceil - 1 \tag{3.30}$$

$$\forall v_i \in V, \sum S_i \cdot (y_i - a_i) \leq \lceil \frac{N}{2} \rceil - 1 \tag{3.31}$$

$$\forall v_i \in V, \sum S_i \cdot (z_i - w_i) \leq \lceil \frac{N}{2} \rceil - 1 \tag{3.32}$$

The objective 3.23 is to minimize the total cost of cut links and disabled DCNs that partitions the network such that no majority for consensus-based algorithms is feasible. Constraints 3.24 and 3.25 guarantee that only DCNs in partition **X** can be disabled. If both x_i and u_i are equal to 1, q_i is equal to 1. Constraints 3.26 and 3.27 state the same for partition **Y**, and constraints 3.28 and 3.29 for **Z**. Finally, constraints 3.30 to 3.32 ensure that the number of active DCNs per partition does not exceed $\lceil \frac{N}{2} \rceil - 1$.

3.7 Model Construction

The modified code for the ILP formulation of the CANLA problem is as follows.

```
N=5
c=10
t=10
#We add the new variables
delta_ij = nw_mod.addVars(V.number_of_nodes(),V.number_of_nodes(),name='delta_{i}
    ↪ _{j}',vtype=GRB.BINARY)
x=nw_mod.addVars(V.number_of_nodes(),name='x',vtype=GRB.BINARY)
y=nw_mod.addVars(V.number_of_nodes(),name='y',vtype=GRB.BINARY)
z=nw_mod.addVars(V.number_of_nodes(),name='z',vtype=GRB.BINARY)
```

```
u=nw_mod.addVars(V.number_of_nodes(),name='u',vtype=GRB.BINARY)
a=nw_mod.addVars(V.number_of_nodes(),name='a',vtype=GRB.BINARY)
b=nw_mod.addVars(V.number_of_nodes(),name='b',vtype=GRB.BINARY)
w=nw_mod.addVars(V.number_of_nodes(),name='w',vtype=GRB.BINARY)
#We update the objective function
obj_fun=sum(c*delta_ij[i,j] for i,j in V.edges())+sum(t*u[k] for k in V.nodes())
nw_mod.setObjective(obj_fun, GRB.MINIMIZE)
#we add the new constraints
for i in V.nodes():
    nw_mod.addConstr(x[i] + y[i] + z[i] == 1, 'constraint_one_subd{i}')

for i in V.nodes():
    nw_mod.addConstr(2*a[i]<=u[i]+x[i],'constraint_ui_xi')
    nw_mod.addConstr(2*b[i]<=u[i]+y[i],'constraint_ui_yi')
    nw_mod.addConstr(2*w[i]<=u[i]+z[i],'constraint_ui_zi')

    nw_mod.addConstr(1+2*a[i]>=u[i]+x[i],'constraint_ui_xi_2')
    nw_mod.addConstr(1+2*b[i]>=u[i]+y[i],'constraint_ui_yi_2')
    nw_mod.addConstr(1+2*w[i]>=u[i]+z[i],'constraint_ui_zi_2')
```

After the updated model is executed, an output of the new objective function value and the variable values like in the illustrative example are observed:

```
Optimization is done. Objective function value: 30.00
List of connections brought down:
Empty DataFrame
Columns: []
Index: []
List of nodes with DCN service brought down:
       0
u[1] 1.0
u[4] 1.0
u[9] 1.0
Nodes in partition x:
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
Nodes in partition y:
[]
Nodes in partition z:
[]
```

The capability for the model to perform a direct interruption of the DCN services on the DCNs under this conditions for c_{ij} and t_i allows for not only a lower objective function value, but also for it no longer needing to distribute active DCNs into several partitions to achieve the constraint of $\leq \lceil \frac{N}{2} \rceil - 1$ DCNs per partition as visualized in Fig. 3.15.
For $t_i > c_{ij}$ and [$t_i = 15$,$c_{ij} = 10$] The optimal solution of model in the illustrative example with $N = 5$ is as follows:

```
Optimization is done. Objective function value: 35.00
List of links cut:
                     0
delta_{i}_{j}[3,8] 1.0
delta_{i}_{j}[6,9] 1.0
List of nodes with DCN service disrupted:
       0
u[4] 1.0
```

```
Nodes in partition X:
[8, 9, 10, 11, 12]
Nodes in partition Y:
[]
Nodes in partition Z:
[0, 1, 2, 3, 4, 5, 6, 7]
```

The change in cost related to the DCN service disabling, t_i, reflects a reconfiguration in the type of result which is optimal. The hybrid attack under these new conditions outputs a hybrid optimal solution, which includes two cuts and one DCN disruption, as visualized in Fig. 3.16. From the result readings, the remaining active DCNs are distributed into two partitions, giving an objective function value of 35 [2 link-cuts and 1 DCN disabled], which is lower than the optimal value in the link-cut only scenario (40 [4 link cuts]).
When $t_i >> c_{ij}$ and [$t_i = 30$,$c_{ij} = 10$], the optimal solution is as follows.

```
Optimization is done. Objective function value: 40.00

List of links cut:
                      0
delta_{i}_{j}[1,4] 1.0
delta_{i}_{j}[3,6] 1.0
delta_{i}_{j}[3,8] 1.0
delta_{i}_{j}[6,9] 1.0
List of nodes with DCN service disrupted:
Empty DataFrame
Columns: []
Index: []
Nodes in partition X:
[0, 1, 2, 3]
Nodes in partition Y:
[4, 5, 6, 7]
Nodes in partition Z:
[8, 9, 10, 11, 12]
```

This iteration repeats the solution visualization of Fig. 3.2 for the illustrative graph. When $t_i >> c_{ij}$, it is highly likely that the model will produce solutions as in instances where only link cuts are used. In other words, in a hybrid attack vector case where disrupting a node is several times more costly than cutting a link for an optimal solution of the CANLA, cutting links will be preferred and the amount of links cut will always be bigger than zero, whilst the number of nodes disrupted will have a tendency towards zero.

3.8 Performance Assessment

We evaluate the performance of the ILP formulation of the CANLA problem on the Atlanta network graph. We set $N = 5$ and consider the three scenarios:

1. $t_i = c_{ij}$
2. $t_i > c_{ij}$
3. $t_i >> c_{ij}$

Fig. 3.17 shows the topology with the DCNs selected arbitrarily, denoted with white: 1, 3, 10, 13, and 14.

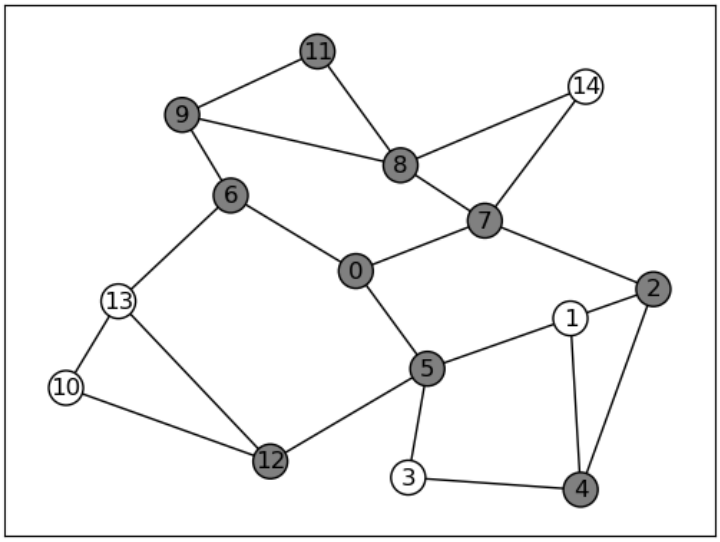


Figure 3.17: The Atlanta graph with indices instead of tags and arbitrarily selected five DCNs.

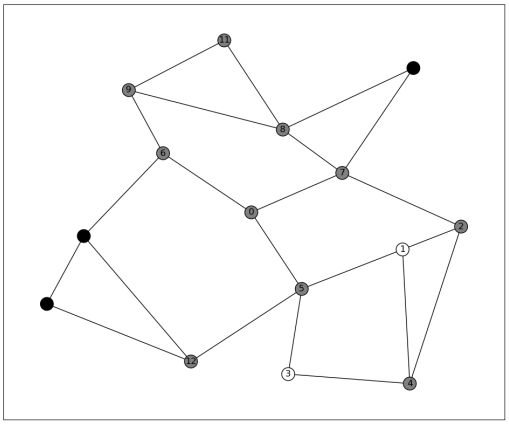
Taking into consideration what was observed from the illustrative model for this conditions, it is plausible to hypothesize that the behaviour for this cost relation can be represented by the mathematical nuance: $t_i = c_{ij} \xrightarrow{likely} \sum H_q >> \sum R_q$, where H_q is the occurrence of DCNs disabling in the placement iteration q (with $H_q = 1$ if there are DCN disabling present in the iteration q with q ranging from 1 to the total number of possible placements and $H_q = 0$ if not), and R_q, which is the occurrence of link cuts in the same placement iteration (with $R_q = 1$ if there is a link cut present in the iteration q with q in the same range and $R_q = 0$ if not); for this value of N.

```
Optimization is done. Objective function value: 30.00
List of connections brought down:
Empty DataFrame

Columns: []
Index: []
List of nodes with DCN service brought down:
        0
u[10] 1.0


u[13] 1.0
u[14] 1.0
Nodes in partition x:
[]
DCN nodes in this partition
Nodes in partition y:
[]
DCN nodes in this partition
```

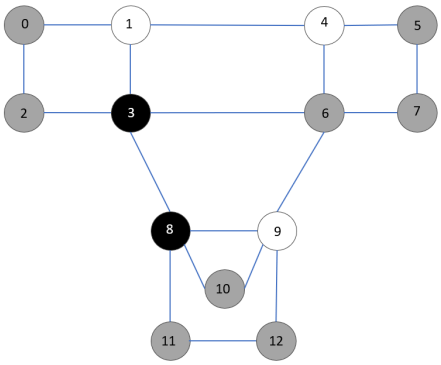
```
Nodes in partition z:
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]
DCN nodes in this partition
```

```
1 N2
3 N4
10 N11
13 N14
14 N15
```

Confirming the assumptions related to the mathematical nuances derived from the illustrative model, the model when having $c_{ij} = t_i$ creates an optimization where $\lceil \frac{N}{2} \rceil + 1$ DCNs are disabled, hence achieving the overall goal without necessarily cutting links, and per consequence not incurring in partitioning at all, just like it occurred with the illustrative model as it can be visualized with Fig. 3.18.



(a) An example result for the hybrid attack against the Atlanta network with a preference for a node-disabling.



(b) The illustrative example network divided into one partition with a three node disabling, cost-minimal, hybrid approach.

Figure 3.18: Comparison of the optimal solutions for the hybrid attack scenario with $t_i = c_{ij} = 10$ for the Atlanta graph and the illustrative graph.

However, to verify the validity of the mathematical statement

$$t_i = c_{ij} \xrightarrow{likely} \sum H_q >> \sum R_q, \tag{3.33}$$

we need to consider all DCN placement combinations and observe how many other minima exist for $t_i = c_{ij} = 10$. All possible values for all combinations are provided in Fig. 3.19, while Fig. 3.20 reports the number of occurrences for DCN disabling and link-cuts are reported. This type of visualization was chosen to improve readability of the cost distribution per DCN placement realization. A rolling average with a window of 50 placements was applied to both the link cost (c,δ) and the DCN disable cost (c,u).

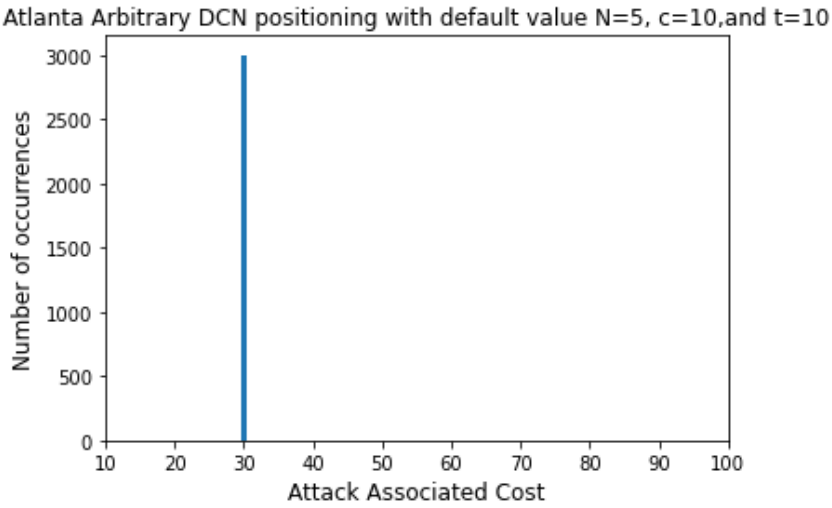


Figure 3.19: The distribution of the total attack budget over all possible DCN placements for the Atlanta graph and N=5 when $c_{ij} = t_i = 10$.

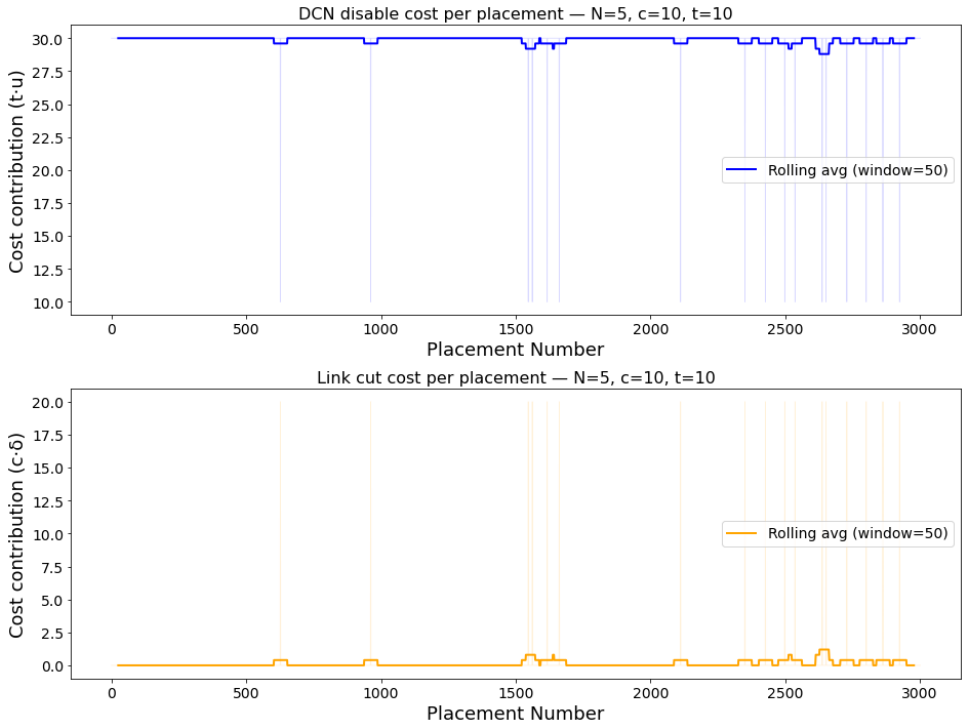


Figure 3.20: Fluctuation of attack choices for the hybrid attack case with $t_i = c_{ij} = 10$. for the Atlanta graph.

With a 100% of occurrence rate observable in figure 3.19, there is only one minimum for all possible DCN distributions, which is 30. The mathematical sense given in equation 3.34 holds to be true not only from the inspection of 3.20, where there are no cases with the cost contribution of cut links exceeding the one of disabled nodes. It is also verifiable by querying the arrays holding the results for cut links and disabled nodes for all iterations, where we can count for how many placements were there zero values associated to each type of attack:

- *Total number of placements with zero cost contributions from disabled DCNs: 0*
- *Total number of placements with zero cost contributions from cut links: 2985*

This can be reinterpreted as the following, with the mathematical nuance shown before in 3.34:

$$\sum q = 3003$$

$$\sum H_q = \sum q - 0$$

$$\sum R_q = \sum q - 2985$$

$$\sum H_q = 3003; \sum R_q = 18$$

$$3003 >> 18$$

Hence, 3.34 holds to be true for this assessment.

We continue with the assessment of the $t_i > c_{ij}$ case on the Atlanta graph. The output obtained after solving the model, showing the objective function value, list of links cut, nodes disabled, and partitioning is as follows.

```
Optimization is done. Objective function value: 35.00
List of connections brought down:
                              0
delta_{i}_{j}[5,12] 1.0
delta_{i}_{j}[6,13] 1.0
List of nodes with DCN service brought down:
          0
u[1] 1.0
Nodes in partition x:
[]
DCN nodes in this partition
Nodes in partition y:
[10, 12, 13]
DCN nodes in this partition
10 N11
13 N14
Nodes in partition z:
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 14]
DCN nodes in this partition
1 N2
3 N4
14 N15
```

In the $t_i > c_{ij}$ case, we observe behaviour similar to the one for the illustrative model: a hybrid choice between both attacks, leveraging the DCN disabling capabilities to eliminate one node, and cut two links that create only two partitions, as visualized in Fig. 3.21.

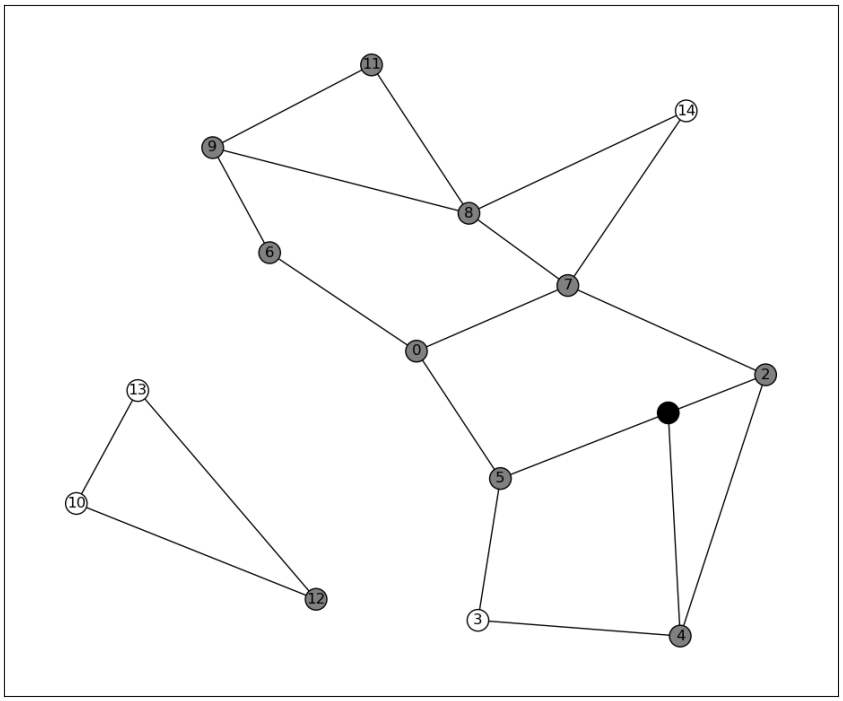


Figure 3.21: An example result for the hybrid attack against the Atlanta network with a preference for link-cuts and node disabling.

The probable mathematical nuance for this case drawn from the observations in the model would be:

$$t_i > c_{ij} \xrightarrow{likely} \sum H_q \geq \sum R_q \tag{3.34}$$

To investigate the validity of this statement, we check the results for all DCN placements. Fig. 3.22 shows the objective function values for each placement, and Fig. 3.23 depicts the number of link cuts and disabled DCN per placement with the same rolling average window for readability purposes.

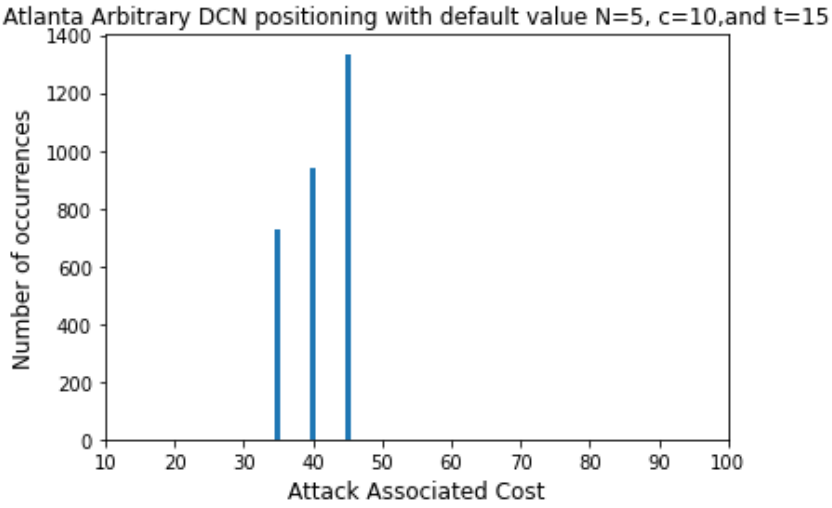


Figure 3.22: The distribution of total attack budget over all possible DCN placements for the Atlanta graph and N=5 when $c_{ij} = 10$ and $t_i = 15$.

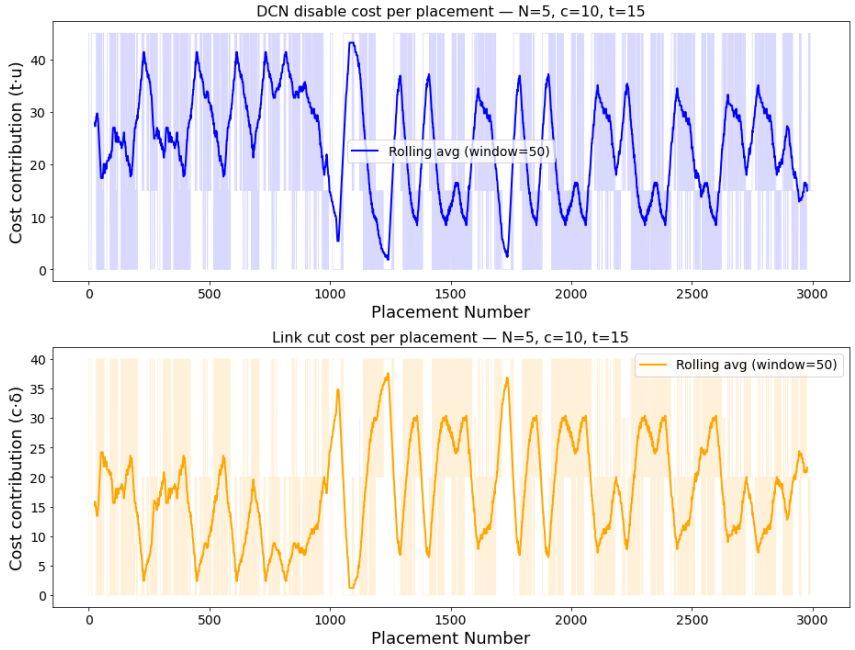


Figure 3.23: Fluctuation of attack choices for the hybrid attack case with $t_i = 15, c_{ij} = 10$ for the Atlanta graph.

The solutions for all arbitrary placements in the $t_i > c_{ij}$ case show that the variation in t_i alters the choice of attacks and provides better cost effectiveness compared to the link-cut only scenario. From Fig. 3.23, one can draw little conclusions from the underlying bar graph. From the rolling window overlying graph we can observe an inverse correlation between the two contribution graphs. However, the fluctuations in both attack contributions makes it hard to observe the preferred attack vector across all the different placements. Better insight can be drawn through numerical analysis of the obtained solution, indicating:

- *Total number of placements with zero cost contributions from disabled DCNs: 942*
- *Total number of placements with zero contributions from cut links: 1322*

Further examination gives:

$$\sum q = 3003$$

$$\sum H_q = \sum q - 942$$

$$\sum R_q = \sum q - 1322$$

$$\sum H_q = 2061; \sum R_q = 1681$$

$$2061 >> 1681$$

This indicates that $\sum H_q \geq \sum R_q$ for $t_i > c_{ij}$ holds to be true for this assessment. Finally, we assess the scenario where $t_i >> c_{ij}$. The considerations for this case are consistent with the ones for the illustrative model: $N = 5$, $c_{ij} = 10$, and $t_i = 30$.

We first show the result for this example run by listing the cut links and disabled DCNs, as well as the objective function value and the partition distribution.

```
 Optimization is done. Objective function value: 40.00
List of connections brought down:
                      0
delta_{i}_{j}[5,12] 1.0
delta_{i}_{j}[6,13] 1.0
delta_{i}_{j}[7,14] 1.0
delta_{i}_{j}[8,14] 1.0
List of nodes with DCN service brought down:
Empty DataFrame
Columns: []
Index: []
Nodes in partition x:
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 11]
DCN nodes in this partition
1 N2
3 N4
Nodes in partition y:
[14]
DCN nodes in this partition
14 N15
Nodes in partition z:
[10, 12, 13]
DCN nodes in this partition
10 N11
13 N14
```

Here, the model deems as the optimal solution to recur to only link cuts, creating three partitions and guaranteeing that the $\leq \lceil \frac{N}{2} \rceil - 1$ DCN condition is satisfied for all three partitions, shown in Fig. 3.24.

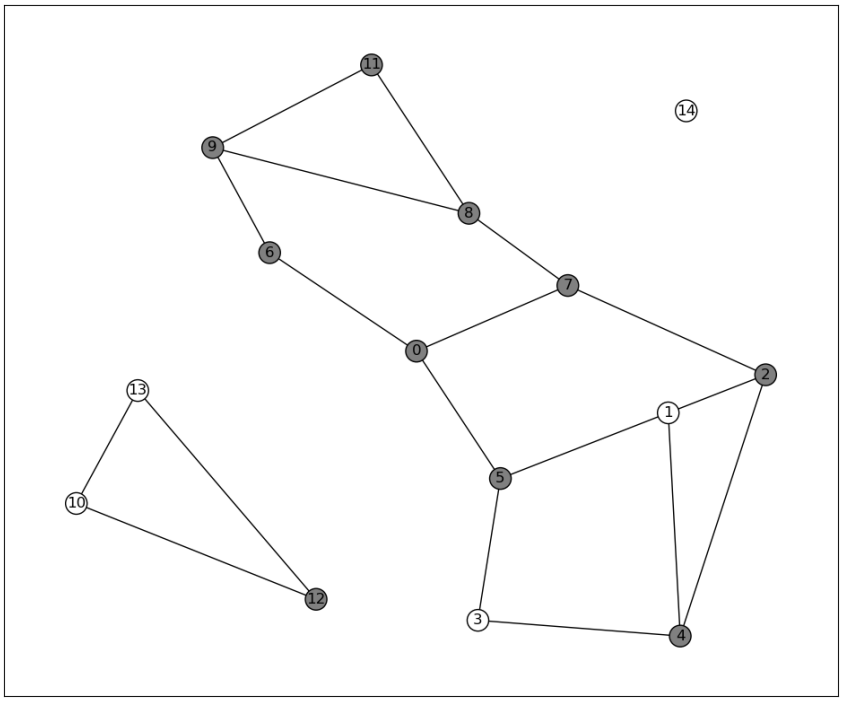


Figure 3.24: An example result for the hybrid attack against the Atlanta network with a preference for link-cuts only.

Fig. 3.25 presents all the objective function values for all possible DCN placement combinations and Fig. 3.26 shows the occurrence fluctuation for link-cuts and node disabling.

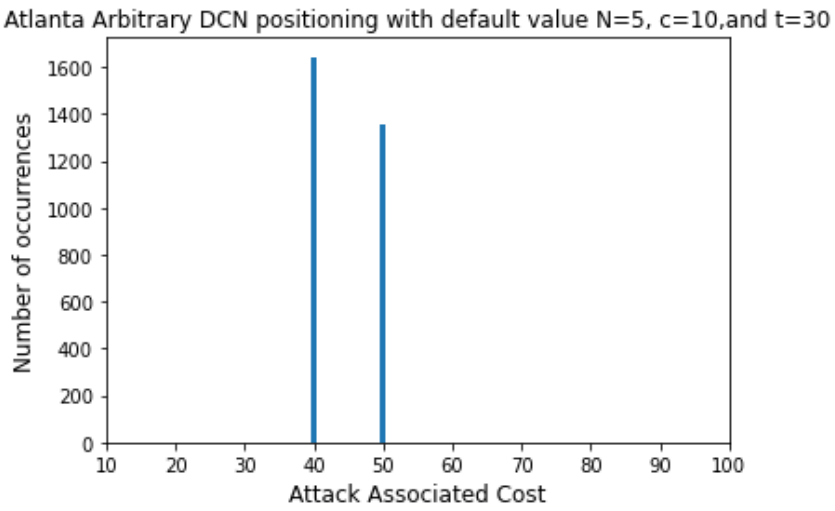


Figure 3.25: Associated weight values obtained for all possible DCN placements (3003) for the Atlanta graph with $t_i = 30, c_{ij} = 10$ and $N = 5$.

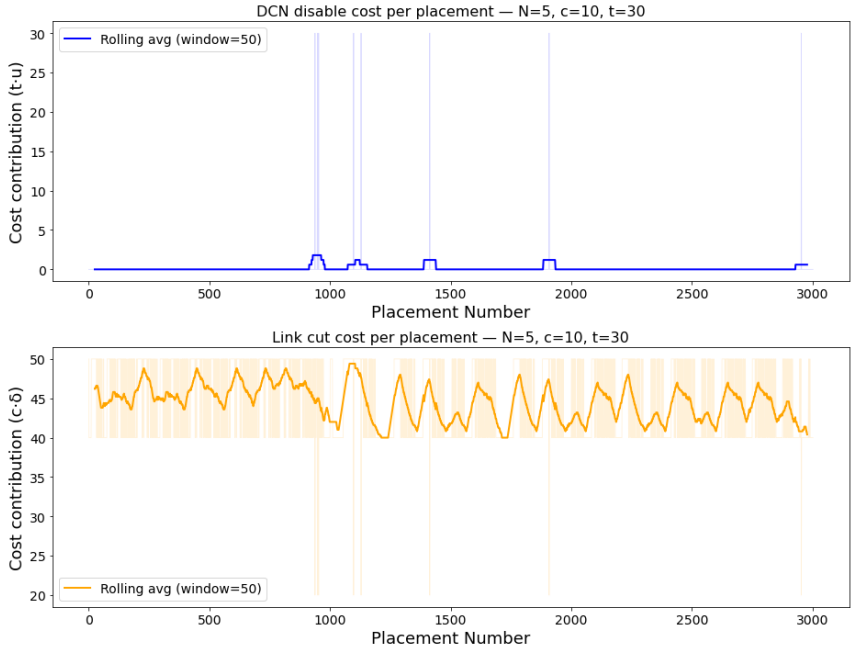


Figure 3.26: Fluctuation of attack choices for the hybrid attack case with $t_i = 30, c_{ij} = 10$ for the Atlanta graph.

For all arbitrary placement combinations in the $t_i >> c_{ij}$ case, the increase in t_i alters the choice of attacks towards the link-cut only scenario, as illustrated in the figures. Mathematically, we can state that $\sum H_q << \sum R_q$ for $t_i >> c_{ij}$. This can be reassessed from Fig. 3.26 as the following:

- *Total number of placements with zero cost contributions from disabled DCNs: 2993*
- *Total number of placements with zero cost contributions from cut links: 0*

With this translating into $\sum H_q = 10$ and $\sum R_q = 3003$, it is demonstrated that $\sum H_q << \sum R_q$ for $t_i >> c_{ij}$ holds to be true for this assessment of this final scenario with the set number of N.

4

Impact of Node Placement on Network Resilience

The use of arbitrary methods for placing DCNs presents significant challenges, particularly due to the vast number of possible configurations that must be considered for large graphs (e.g., the US-CA graph discussed earlier). Additionally, determining the optimal number of DCNs for each scenario remains a critical issue. This chapter introduces a more efficient, structure-aware approach that leverages the inherent properties of graphs. We analyze the impact of the number and placement of DCNs on network resilience. This analysis helps us to identify the more effective node placement strategies as well as the corresponding number of DCNs.

4.1 Structural node features for DCN placement

Policies for selecting the locations for DCNs can be guided by various metrics. These can rely on properties that are intrinsic to the network elements and represent the significance of the nodes or links. The importance of a node within a graph can be described by, among others, node degree centrality, node closeness centrality, or node betweeness centrality, metrics that gauge the structural importance and connectivity of nodes within the graph [21].

Node degree centrality

The degree centrality of node i is equal to the sum of the weights of all links connecting node i with its neighbors, expressed as $deg(i) = \sum_{j \in N(i)} w_{ij}$. In the case of unitary link weights, such expression is reduced to the number of directly connected neighbors of a node. The more directly connected neighbors a node has, for a unitary weighted link graph, the more central it is.

Node closeness centrality

Node closeness centrality evaluates how close a node is on average to all other nodes within the network. By measuring the length of the shortest path from every other node to node i, its closeness centrality is expressed as $clo(i) = \sum_{j \in N->i} \frac{w_{ij}}{d_{ij}}$. In the case of unitary link weights, as considered in this project, the expression is reduced to $clo(i) = \sum_{j \in N->i} \frac{1}{d_{ij}}$. The more central a node is, the more important it is to the functioning of the network.

When selecting the placement of DCNs as an input to the resilience evaluation, we use the degree and closeness centrality and investigate four strategies:

- High Degree Centrality (HDC)
- Low Degree Centrality (LDC)
- High Closeness Centrality (HCC)
- Low Closeness Centrality (LCC)

In each of them, the nodes are sorted in the descending (or the ascending) order according to their respective centrality feature. After that, the first N nodes from the sorted list are selected to host DCNs. We analyze and compare these four non-arbitrary methods against the average performance of the arbitrary one for the different numbers of DCNs and different attack scenarios in the next section.

4.2 Node-Feature-Aware Selective Placement vs. Arbitrary Placement

4.2.1 Link-cut only attacks

We first consider the link-only attack scenario and evaluate the performance of the CALA problem for the different DCN placement strategies. In all reported results, $N = 2k + 1$, where k takes values from $k = 1...\frac{|V|-1}{2}$ for graphs with an odd $|V|$ and $k = 1...\frac{|V|-2}{2}$ for graphs with an even $|V|$. For the EU and the US-CA graph, the average objective function value shown for the different numbers of N is taken from a sample of 5000 of all possible placements of N DCNs when this combination number is too high. This is due to the fact that the number of possible placements is $\binom{|V|}{N}$.

4.2.1.1 Atlanta graph

Fig. 4.1 shows the performance for the four placement methods against the average for arbitrary placements for different values of N in the Atlanta graph. As can be seen in the figure, HDC and HCC are more effective than LCC or LDC in enhancing network resilience against this specific attack model. This is because HDC selects nodes with the most connections, making them more difficult to isolate. Similarly, HCC prioritizes nodes that lie on many shortest paths, which include mostly centrally located nodes. Another observation can be made from the results: an increase in the number of DCNs does not necessarily improve network resilience. On the contrary, the performance stagnates. This can be explained by the fact that a larger number of DCNs requires a larger number of nodes to form a majority and reach consensus. Attacks can exploit this requirement and partition the network with fewer cuts than for a DCN infrastructure with fewer DCNs.

4.2.1.2 EU graph

Fig. 4.2 shows the performance for the four placement methods against the average arbitrary placement approach for different values of N in the EU graph. For cases in which the value of N produces a very large number of combinations, the results

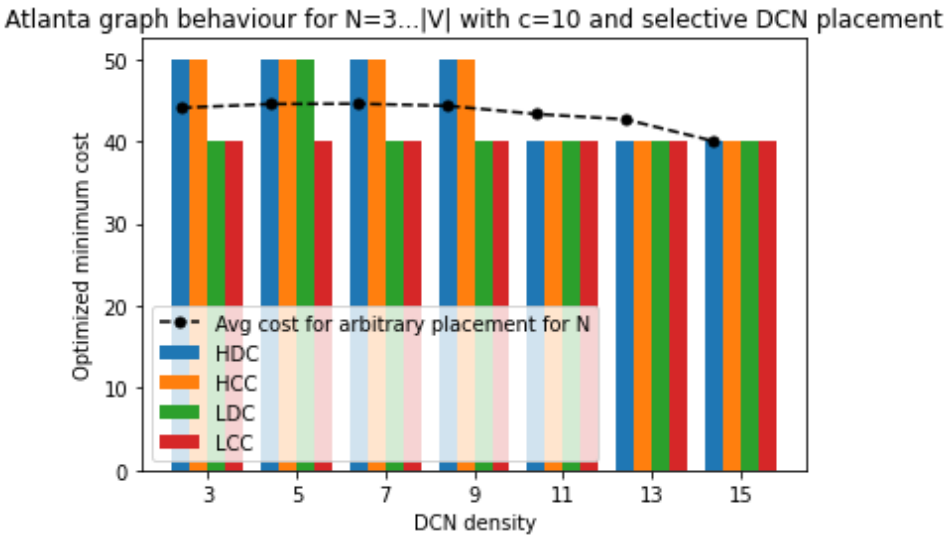


Figure 4.1: Attack budget for the Atlanta topology for four methods with $c_{ij} = 10$ against average cost of arbitrary placement, for different number of DCNs.

represent an average over a sample of 5000 arbitrary non-repeated placements. The number of possible combinations for all N is provided in Table 4.1.

N	Num. Poss.	N	Num. Poss.
3	3276	17	21474180
5	98280	19	6906900
7	1184040	21	1184040
9	6906900	23	98280
11	21474180	25	3276
13	37442160	27	28
15	37442160		

Table 4.1: The number of possible arbitrary placement combinations for the EU graph with $|V| = 28$.

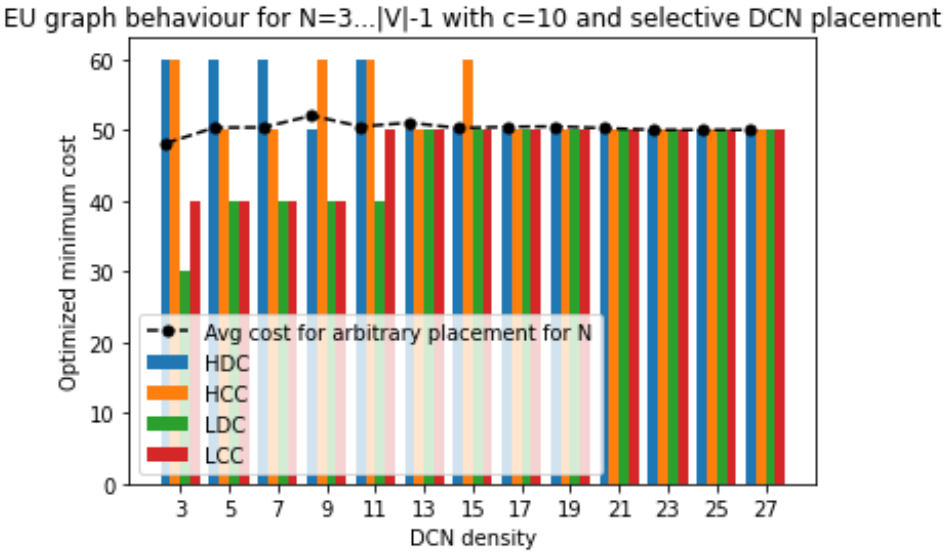


Figure 4.2: Attack budget for the EU topology for four placement methods with $c_{ij} = 10$ against cost of arbitrary placement, for different number of DCNs.

4.2.1.3 US-CA graph

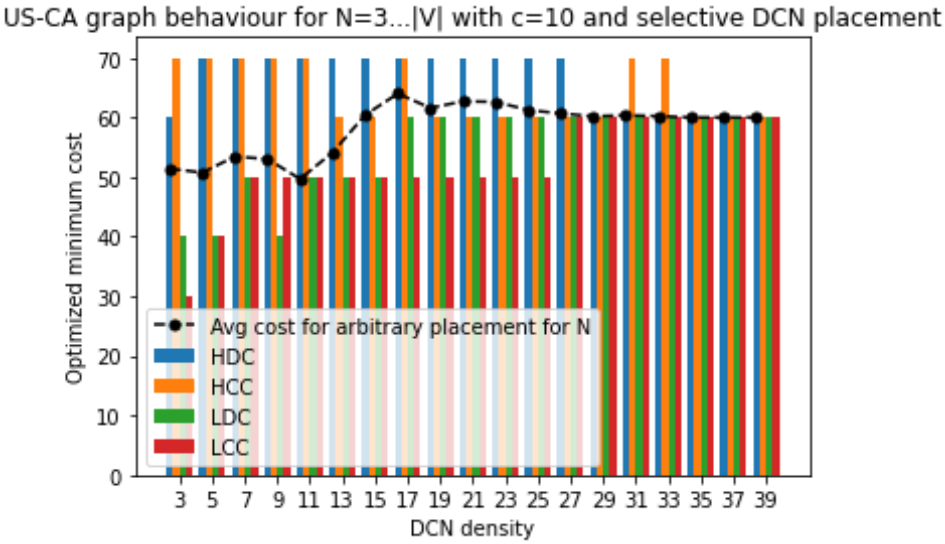


Figure 4.3: Attack budget for the US-CA topology for four methods with $c_{ij} = 10$ against cost of arbitrary placement, for different number of DCNs.

Fig. 4.3 shows the results for the US-CA network topology. The results indicate that the HDC and HCC placement policies are more resilient than LCC or LDC. HDC performs best for moderate values of N (e.g. $N = 21$), and HCC for higher values of N (e.g. $N = 33$). The stagnation in performance for these approaches begins at much higher densities, e.g., at $N = 29$, and despite the surges in the HCC performance for $N = 31$ and $N = 33$, it stabilizes at $N = 35$.
Regarding the computation time, the calculations necessary for the centrality metrics placements took less than a minute. When calculating all possible placements for $N = 5$ only for the US-CA graph, the model took around 14.5 hours. With this

taken into consideration, for DCNs $N = 19$, it would take 1,570,467.38 hours or 65,436.14 days.

4.2.1.4 Discussion

There are important remarks to mention after analysing all the results as a whole from the experimentation carried out in this section:

- The more is not always the better. As the results reflect, there is a sweet spot where with a moderate number of DCNs, the model attains an upper bound on the attack resilience, and the performance decreases if the number of DCNs continues growing.
- Stagnation occurs in all cases. With a growing number of DCNs, the non-arbitrary well-performing placement methods stop outperforming the average cost for their particular value of N. For values of N that approach $|V|$, the greedy search of placements that give values closer to the upper bound could be a better fit.
- In a fast-paced and time-dependent endeavor like network and communications are, the need for quickly easily obtainable solutions arises. Hence, a selective model as the proposed one which takes very small amounts of time regardless of the graph size could be considered a serious alternative to an extensive all-possibilities approach, which has shown to consume higher amounts of time as the graph grows.

4.2.2 Hybrid Attacks

This section compares the performance of selective DCN placement and average arbitrary placement when addressing the CANLA problem. The analysis employs the same four selection methods and cost variations introduced in the hybrid attack model section for the Atlanta graph.

4.2.2.1 The $t_i = c_{ij}$ case

Fig. 4.4 shows the performance for the four placement methods against the average arbitrary placement approach for different values of N in the Atlanta graph for the hybrid attack approach with $t_i = c_{ij} = 10$.

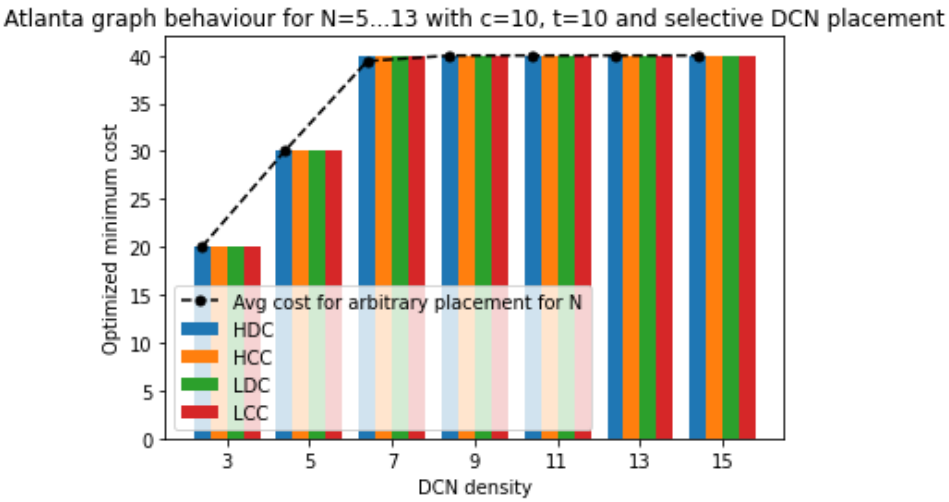


Figure 4.4: Attack budget for Atlanta graph for different number of DCNs and four methods with $c_{ij} = t_i = 10$

The selective method does not perform well, as from Fig. 4.4 it becomes notable that for all values of N the four methods reach the average objective value for all arbitrary placements.

Another key aspect we aimed to establish was the mathematical insights derived for $N = 5$ in the previous chapter. To achieve this, Fig. 4.5 presents a comparison of the average number of disabled DCNs and cut links in the Atlanta graph, using both arbitrary and non-arbitrary selection methods.

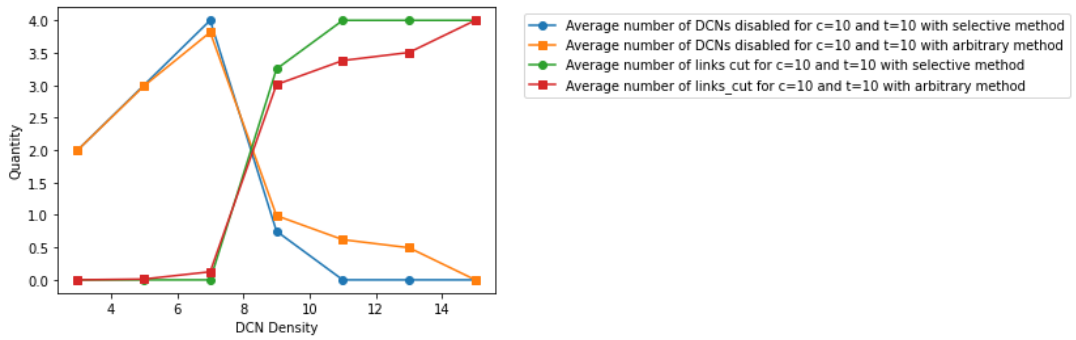


Figure 4.5: Average cost of cut links and disabled nodes for Atlanta graph for different DCN densities with $c_{ij} = t_i = 10$.

For $N = 5$ with $t_i = c_{ij} = 10$ case, the following had been established:

$$t_i = c_{ij} \xrightarrow{likely} \sum H_q >> \sum R_q \tag{4.1}$$

By inspecting the fluctuation of the average quantities for δ_{ij} and u_i through N with this value for c_{ij} and t_i, a more uniform approach can be settled: If one considers an equal cost model for the values associated to the link cuts (δ_{ij}) and for the node disruption (u_i), when increasing the value of N approaching $|V|$, there is an inflection point for the model where the choice for link cuts overcomes the choice for node disruption when approaching a particular N for this graph. From Fig. 4.5 it can be assessed that such inflection happens around the value of $N \approx 8$. In mathematical terms, this could translate as the following approximation for the Atlanta graph:

$$f(H_q, R_q, t_i = c_{ij}) \begin{cases} \sum H_q >> \sum R_q & \text{if } \lim_{N^- \to 8}, \\ \sum H_q << \sum R_q & \text{if } \lim_{N^+ \to 8}. \end{cases} \tag{4.2}$$

This observation can be attributed to the fact that when we increase the value of N, the limit of DCNs that can be cut also approaches the value of δ_{ij} that is optimal for the cut only scenario for the same N. Given the fact that the model will always find the optimal lowest cost solution, this trend change in attack vector preference presents itself as shown.

4.2.2.2 The $t_i > c_{ij}$ case

Fig. 4.6 shows the performance for the four placement methods against the average arbitrary placement approach for different values of N in the Atlanta graph for the hybrid attack approach with $t_i = 15$ and $c_{ij} = 10$.

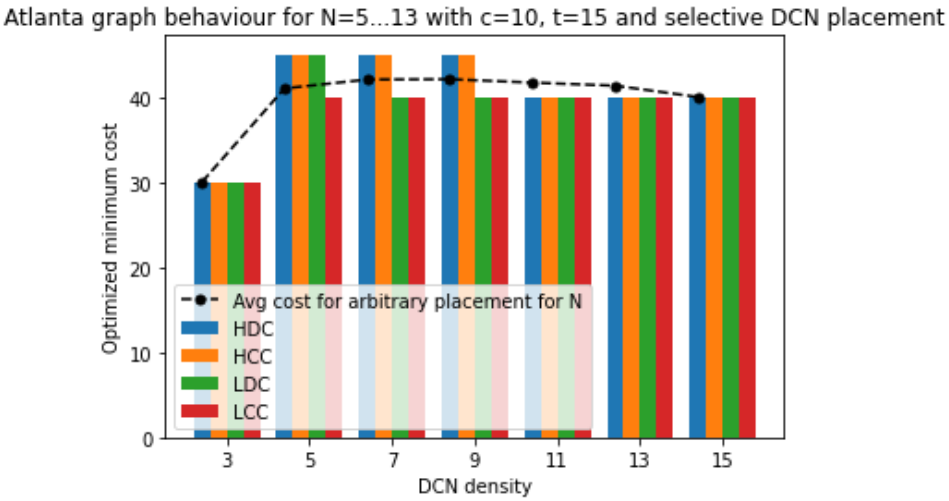


Figure 4.6: Attack budget for Atlanta graph for different number of DCNs and four placement methods with $c_{ij} = 10$ and $t_i = 15$.

In the case of $t_i > c_{ij}$, a good performance is observed for the selective methods HDC and HCC, similarly to the link-cut only case, as is the stagnation at bigger numbers of N. This could mean that the model behaves similarly with both link-cut only and hybrid attack vectors when $c_{ij} < t_i$. Same as with the previous cost scenario, another aspect that we were looking to establish was the mathematical nuance obtained for $N = 5$ in the previous chapter. To this end, Fig. 4.7 shows a comparison between the average cost of disabled DCNs cut and links in the Atlanta graph for both arbitrary and non-arbitrary methods.

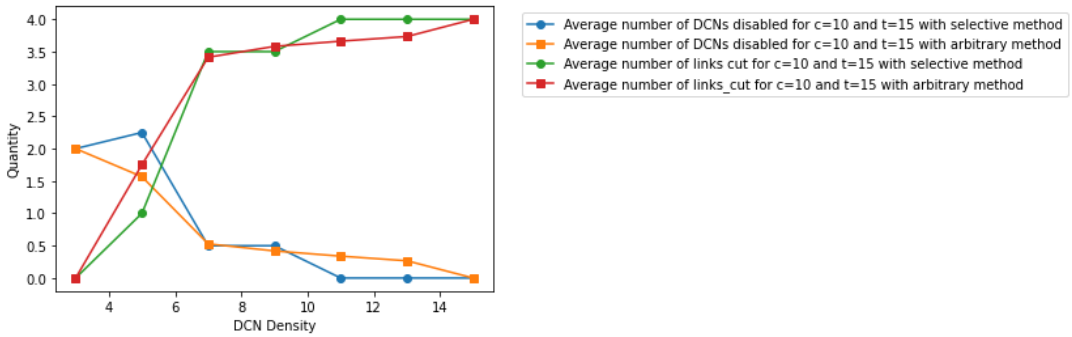


Figure 4.7: Average link-cuts and disabled nodes for Atlanta graph for different DCN densities with $c_{ij} = 10$ and $t_i = 15$.

In a similar fashion, by inspecting the fluctuation of the average quantities for δ_{ij} and u_i through N with this value for c_{ij} and t_i, a more uniform approach can be settled: If one considers higher cost values associated to the link cuts (δ_{ij}) than for the node disruption (u_i), when incrementing the value of N approaching $|V|$, there is a constant increase in the average amount of links cut until the value of approximately $N = 5$. After reaching this value of N, the growth stagnates into a very slowly decreasing function for node disruptions and a very slowly increasing function for link cuts. In mathematical terms, this could translate as the following

approximations:

$$f(H_q, R_q, t_i > c_{ij}) \begin{cases} \sum H_q > \sum R_q & \text{if } \lim_{N^- \to 5}, \\ \sum H_q < \sum R_q & \text{if } \lim_{N^+ \to 5}. \end{cases} \tag{4.3}$$

This observation can be attributed to the fact that when we increase the value of N with this values for t_i and c_{ij}, the number of DCNs that can be disabled for an optimal solution represent a higher incur in cost than with link cuts for the same N. Since in this case the cost is just slightly different (50%), the observed change is slowly envisioned until this critical value is reached.

4.2.2.3 For $t_i = 30$ and $c_{ij} = 10$

Fig. 4.8 shows the performance for the four placement methods against the average arbitrary placement approach for different values of N in the Atlanta graph for the hybrid attack approach with $t_i = 30$ and $c_{ij} = 10$.

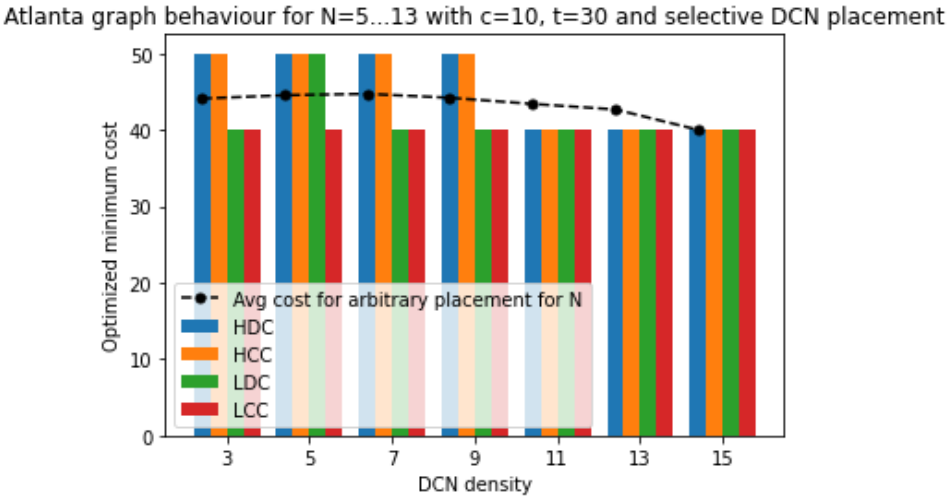


Figure 4.8: Attack budget for Atlanta graph for different number of DCNs and four methods with $c_{ij} = 10$ and $t_i = 30$

In the case of $t_i >> c_{ij}$, a good performance is observed for the selective methods HDC and HCC, similarly to the link-cut only case and with the previous scenario where $c_{ij} < t_i$, as well as stagnation at larger values of N. This could mean that the model behaves similarly with both link-cut only and hybrid attack vectors when $c_{ij} << t_i$, just as it did with the $c_{ij} < t_i$ scenario. Same as with the first two cases, another aspect that we were looking to establish was the mathematical nuances obtained for $N = 5$ in the previous chapter. To this end, 4.9 compares the average number of disabled DCNs and cut links in the Atlanta graph for the arbitrary and non-arbitrary methods.

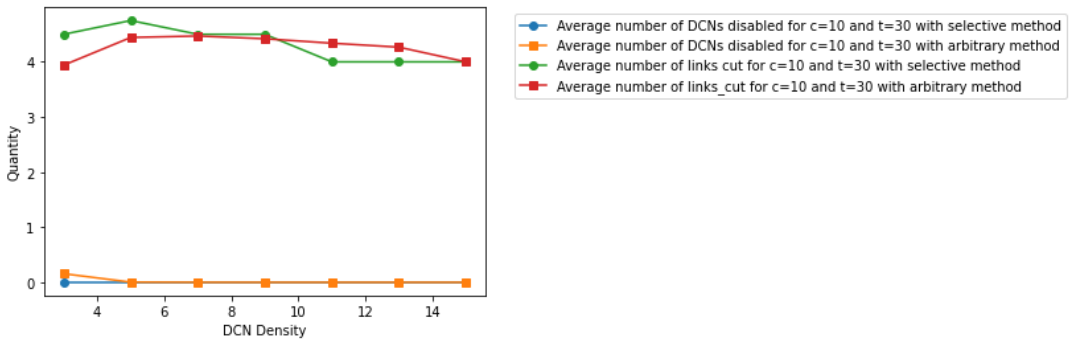


Figure 4.9: Average link-cuts and disabled nodes for Atlanta graph for different DCN densities with $c_{ij} = 10$ and $t_i = 30$.

The figure shows that, in the average placement case, the number of disabled nodes for $c_{ij} << t_i$ is mostly zero or close to it, which makes the model exhibit behaviour characteristic of the link-cut only scenario.This is represented by the following mathematical assertion:

$$f(H_q, R_q, t_i >> c_{ij}) = \sum H_q << \sum R_q \text{ for all } N \tag{4.4}$$

Complementing the two previous cases ($c_{ij} < t_i$ and $c_{ij} = t_i$), it can be concluded that the point that marks the trend change in both functions is transferred to the left as t_i grows with respect to c_{ij}, and vice versa when plotted against N for the relation $c_i j \leq t_i$. This can be interpreted as the translation of the critical point being dependent on the ratio $\frac{c_{ij}}{t_i}$.

However, in order to determine the natural positioning of such critical point along the N axis, a proper assessment is carried out for the other complementary graphs to observe their behaviour to such related cost changes and values of N.

In Figs. 4.10 to 4.12, it is noticeable that there is no phenomenon variation between graphs, but the same behaviour arises around the same values of N, regardless of the graph size and the number of possible values that N can take for each graph.

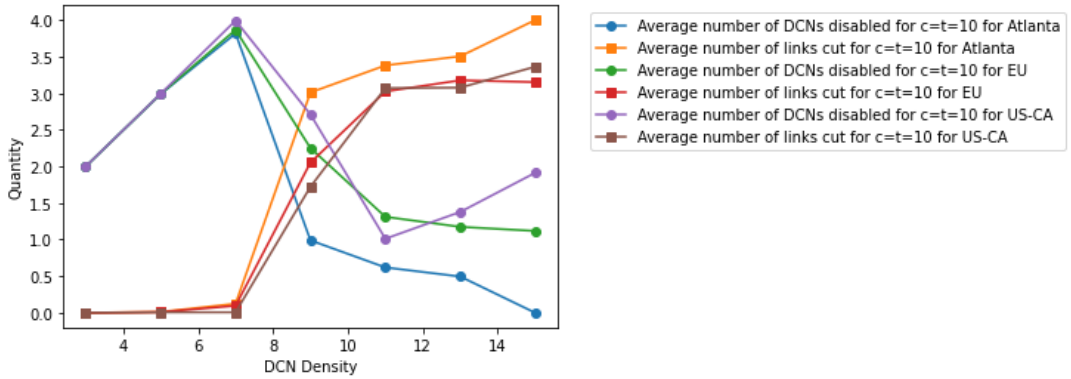


Figure 4.10: Average link-cuts and disabled nodes for all assessment graphs for different number of DCNs with $c_{ij} = t_i = 10$.

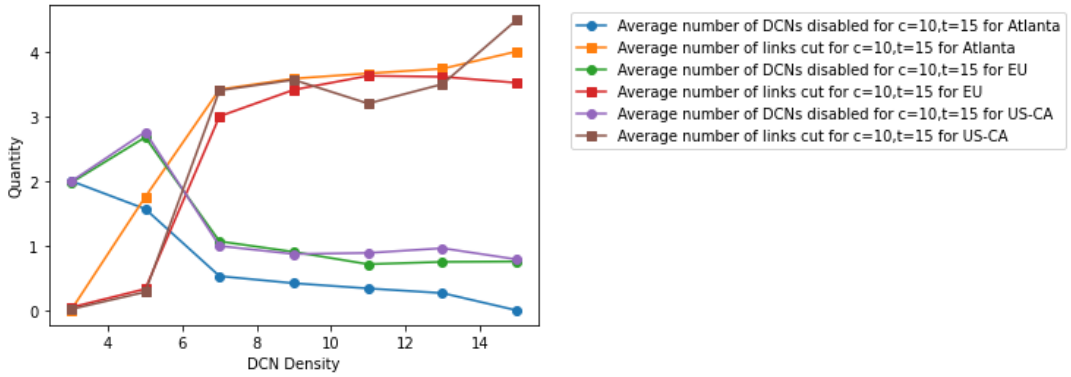


Figure 4.11: Average link-cuts and disabled nodes for all assessment graphs for different number of DCNs with $c_{ij} = 10, t_i = 15$.

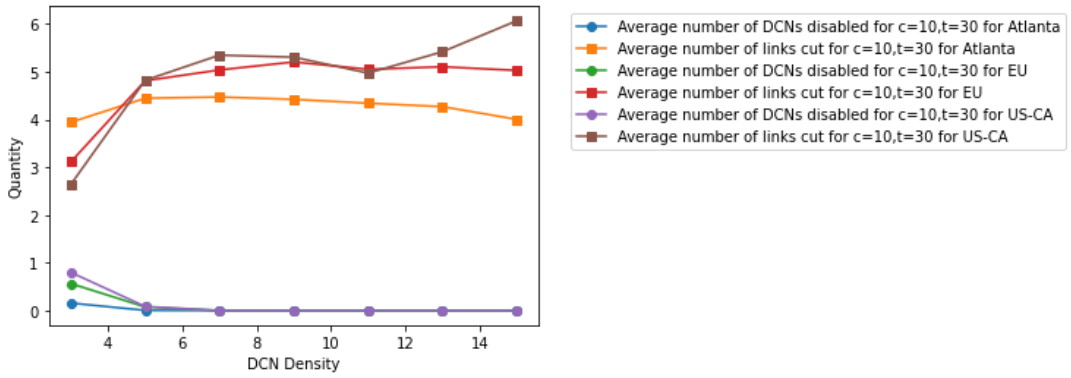


Figure 4.12: Average link-cuts and disabled nodes for all assessment graphs for different number of DCNs with $c_{ij} = 10, t_i = 30$.

The observations from Figs. 4.10 to 4.12 indicate that, for all values that follow the cost model $c_{ij} \leq t_i$, the inflection point for attack vector preference is approximately the same for any graph regardless of the maximum value that N can take, and purely reliant on the cost model for graphs that follow the definitions stated earlier in this work.
For the cost model $c_{ij} \leq t_i$, the transition point is innately around the value where $N \approx 8$ for all performance assessment graphs (meaning when $c_{ij} = t_i$). Such point, which will be defined as $\xi_{\delta u}$, can be approximated as:

$$\xi_{\delta u} \approx 8 \cdot \frac{c_{ij}}{t_i} \tag{4.5}$$

With this definition, it can be approximated that the transition point is located around $\xi_{\delta u} \approx 5.33$ for $t_i = 15$ and $c_{ij} = 10$, and $\xi_{\delta u} \approx 2.66$ for $t_i = 30$ and $c_{ij} = 10$. These two approximations can be verified when looking at Figs. 4.11 and 4.12. With this in mind, for any intermediate values that follow the cost model $c_{ij} \leq t_i$, the $\xi_{\delta u}$ point can be approximated. Around such point the attack vectors are maximized, making the value of N closer to such point the optimal value of N alongside a selective or arbitrary method of DCN placement, potentially narrowing down the optimal N search and saving computing time.

5
Conclusions

The work carried out in this thesis consisted in the formulation and development of an ILP model which provided insights into the susceptibility of distributed data center networks against physical attacks which included link-cuts and node disruption. The outputs of the link-cut only model were presented graphically and analyzed for a particular number of N to obtain understanding of how the optimization takes place. Reference network topologies of different sizes were used for performance assessment in order to verify the functionality of the model for any given network with unitary weight bidirectional optical links. This provided a first insight into how any given circumstance of a graph with a $\binom{|V|}{N}$ number of possible placements has a limited number of minimum link cuts that partition the network and disable the DCN functionalities. A hybrid attack approach was also developed by adding DCN disruption capabilities to the ILP model. The tuning of the associated costs showed revealed fluctuations in optimal attack choices for all placements. A mathematical model for this behaviour was derived to describe how one type of attack is favoured over the other depending of the relationships between costs of disabling links and nodes.

During simulation analysis, a structure-aware selective approach was introduced as an alternative to the exhaustive search for all the possible combinations of DCN placements. This method includes the concepts of HCC, HDC, LCC, and LDC, which all were compared against the average objective function for the different placements of N DCNs. For the cut-only attack approach, we observe that the best performing methods are HCC and HDC, which place DCNs as the most central nodes in the graph. The second observation was that there is a point in all graph in which increasing the number of DCNs does not translate to better performance (stagnation), given that N beyond this value no longer outperforms the average cost for arbitrary placements. For the hybrid approach, we measure the performance against the three considered cost cases to better understand the influence of the values of t_i and c_{ij} against the values of N for attack choices. This provided a better insight into an approximation for N where both link-cuts and node disabling are maximized. The observation allows to set the approximate value $\xi_{\delta u}$ applicable to all graphs that follow the conventions established through this work, enabling a narrower optimal search for the best performing DCN placement for similar graphs and cost model. The models developed in this project provide important insights into the physical security of distributed data center networks and a useful planning tool for maximizing the effort needed for a potential attacker to disrupt consensus-based data center networks.

Bibliography

[1] "Rising cyber threats pose serious concerns for financial stability," 4 2024. [Online]. Available: https://www.imf.org/en/Blogs/Articles/2024/04/09/rising-cyber-threats-pose-serious-concerns-for-financial-stability

[2] C. Natalino, A. de Sousa, L. Wosinska, and M. Furdek, "Content placement in 5g-enabled edge/core data center networks resilient to link cut attacks," *Networks*, vol. 75, no. 4, pp. 392–404, 2020.

[3] Z. N. Rashid, S. R. Zebari, K. H. Sharif, and K. Jacksi, "Distributed cloud computing and distributed parallel computing: A review," in *2018 International Conference on Advanced Science and Engineering (ICOASE)*. IEEE, 2018, pp. 167–172.

[4] A. T. Atieh, "The next generation cloud technologies: a review on distributed cloud, fog and edge computing and their opportunities and challenges," *ResearchBerg Review of Science and Technology*, vol. 1, no. 1, pp. 1–15, 2021.

[5] M. Furdek, "Untitled," *Unpublished*, 2023.

[6] N. L. Shetty Sachin, Kamhoua Charles A., *Blockchain for Distributed Systems Security*. IEEE Computer Society, 2019.

[7] R. Van Renesse and D. Altinbuken, "Paxos made moderately complex," *ACM Computing Surveys (CSUR)*, vol. 47, no. 3, pp. 1–36, 2015.

[8] L. Lamport, "Paxos made simple," *ACM SIGACT News (Distributed Computing Column) 32, 4 (Whole Number 121, December 2001)*, pp. 51–58, 2001.

[9] "What is Data Center Security? | VMware Glossary," 3 2023. [Online]. Available: https://www.vmware.com/topics/glossary/content/data-center-security.html

[10] Chkadmin, "What is Data Center Security?" 5 2022. [Online]. Available: https://www.checkpoint.com/cyber-hub/cyber-security/what-is-data-center/what-is-data-center-security/

[11] "What is data center security?" 2 2024. [Online]. Available: https://www.cisco.com/c/en/us/solutions/security/secure-data-center-solution/what-is-data-center-security.html

[12] "What is Data Center Security? Why is it important? | Fortinet." [Online]. Available: https://www.fortinet.com/resources/cyberglossary/data-center-security

[13] S. S. Rao, *Engineering Optimization - Theory and Practice (5th Edition)*. John Wiley Sons, 2020. [Online]. Available: https://app.knovel.com/hotlink/toc/id:kpEOTPE021/engineering-optimization/engineering-optimization

[14] "Gurobi Optimization." [Online]. Available: https://www.gurobi.com/

[15] "IBM ILOG CPLEX Optimization Studio." [Online]. Available: https://www.ibm.com/products/ilog-cplex-optimization-studio

[16] "GLPK (GNU Linear Programming Kit)." [Online]. Available: https://www.gnu.org/software/glpk/glpk.html,

[17] C. Natalino, A. Yayimli, L. Wosinska, and M. Furdek, "Content accessibility in optical cloud networks under targeted link cuts," in *2017 International Conference on Optical Network Design and Modeling (ONDM)*. IEEE, 2017, pp. 1–6.

[18] ——, "Link addition framework for optical cdns robust to targeted link cut attacks," in *2017 9th International Workshop on Resilient Networks Design and Modeling (RNDM)*. IEEE, 2017, pp. 1–7.

[19] ——, "Infrastructure upgrade framework for content delivery networks robust to targeted attacks," *Optical Switching and Networking*, vol. 31, pp. 202–210, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1573427718300535

[20] S. Orlowski, M. Pióro, A. Tomaszewski, and R. Wessäly, "SNDlib 1.0–Survivable Network Design Library," in *Proceedings of the 3rd International Network Optimization Conference (INOC 2007), Spa, Belgium*, April 2007, http://sndlib.zib.de, extended version accepted in Networks, 2009. [Online]. Available: https://opus4.kobv.de/opus4-zib/frontdoor/deliver/index/docId/958/file/ZR-07-15.pdf

[21] M. Furdek, "Network models and standards," eEN115: Introduction to Communication Networks Course Lecture Slides from Chalmers University of Technology.

A

Appendix 1

Access to the public Github repository:
`https://github.com/rockofut17/Chalmers-Security-Analysis-of-Distributed-Consensus-based-Network-Architecture-ILP`

DEPARTMENT OF ELECTRICAL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY