



HAL
open science

Optimal deployment of virtual network functions for securing telecommunication networks against distributed denial of service attacks: a robust optimization approach

Céline Gicquel, Sonia Vanier, Alexandros Papadimitriou

► To cite this version:

Céline Gicquel, Sonia Vanier, Alexandros Papadimitriou. Optimal deployment of virtual network functions for securing telecommunication networks against distributed denial of service attacks: a robust optimization approach. *Computers and Operations Research*, In press, 105890, https://www.sciencedirect.com/science/article/pii/S0305054822001563?utm_campaign=STMJ_AUTH_SE 10.1016/j.cor.2022.105890 . hal-03647126v1

HAL Id: hal-03647126

<https://hal.science/hal-03647126v1>

Submitted on 20 Apr 2022 (v1), last revised 6 Jun 2022 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Optimal deployment of virtual network functions for securing telecommunication networks against distributed denial of service attacks: a robust optimization approach

Céline Gicquel^a, Sonia Vanier^b, Alexandros Papadimitriou^c

^a*Laboratoire Interdisciplinaire des Sciences du Numérique, Université Paris Saclay, France*

^b*Laboratoire d'Informatique de l'Ecole Polytechnique, Université Paris 1, France*

^c*Orange Labs Products & Services, France*

Abstract

Distributed Denial of Service (DDoS) cyberattacks represent a major security risk for network operators and internet service providers. They thus need to invest in security solutions to protect their network against DDoS attacks. The present work focuses on deploying a network function virtualization based architecture to secure a network against an on-going DDoS attack. We assume that the target, sources and volume of the attack have been identified. However, due to 5G network slicing, the exact routing of the illegitimate flow in the network is not known by the internet service provider. We seek to determine the optimal number and locations of virtual network functions in order to remove all the illegitimate traffic while minimizing the total cost of the activated virtual network functions. We propose a robust optimization framework to solve this problem. The uncertain input parameters correspond to the amount of illegitimate flow on each path connecting an attack source to the target and can take values within a predefined uncertainty set. In order to solve this robust optimization problem, we develop an adversarial approach in which the adversarial sub-problem is solved by a Branch & Price algorithm. The results of our computational experiments, carried out on medium-size randomly generated instances, show that the

Email addresses: celine.gicquel@lri.fr (Céline Gicquel),
vanier@lix.polytechnique.fr (Sonia Vanier)

proposed solution approach is able to provide optimal solutions within short computation times.

Keywords: Telecommunication networks, Cybersecurity, Distributed denial of service, Network function virtualization, Robust optimization, Adversarial approach, Mixed-integer linear programming, Branch & Price, Column Generation

1 Introduction

Distributed Denial of Service (DDoS) attacks are among the top threats to network operators and internet service providers (ISPs). A distributed denial of service is a type of cyberattack in which multiple compromised computer systems attack a target, such as a server or a website, and cause a denial of service for its legitimate users. DDoS flooding attacks are often launched through the use of botnets. A botnet is a network of user computers or Internet of Things (IoT) devices that are remotely controlled by a hacker through malwares. Under the direction of the hacker, an army of botnets can launch a DDoS attack against a target by simultaneously sending to it a large amount of traffic or service requests. The flood of incoming messages, connection requests or malformed packets exhausts the resources of the target and forces it to slow down or even shut down, thereby preventing it to provide service to its legitimate users.

In recent years, the number, intensity and diversity of DDoS attacks have increased dramatically. Thus, in 2016, the BBC website was targeted by a DDoS attack of more than 600 Gbps and was unavailable for a few hours (Khandelwal, 2016). More recently, Amazon announced that its AWS Shield service mitigated a 2.3Tbps DDoS attack in February 2020 (AWS, 2020). There is also a continuous appearance of new attack vectors, i.e. new techniques enabling hackers to launch a DDoS attack, and new combinations of attack vectors: see e.g. the recent report provided in (Netscout Systems, 2020) and (FBI, 2020). This trend is likely to continue and even accentuate in the near future. Namely, with the development of the Internet of Things, systems based on smart devices (such as sensors) connected to the Internet are widely deployed. This increases the vulnerability of networks and the number of potential DDoS targets: see among others Rahimi et al. (2018), Akpakwu et al. (2018), Fysarakis et al. (2016) and Silva et al. (2020). Furthermore, as mentioned e.g. by Grawe (2020), the COVID-19 pandemic

30 has forced organizations to accelerate their digital transformation plans, thus
31 further increasing the attack surface for hackers and criminals.

32 DDoS attacks can be very damaging for the organization they target. For
33 instance, a survey carried out in 2017 by the cybersecurity company Kaspersky
34 Lab estimated the average cost of a DDoS attack for large (1000+) businesses
35 to be around \$2.3 millions (Berard, 2018). This cost mainly comprises the
36 cost incurred in fighting the attack and restoring service, the investment in
37 an offline or back-up system while online services are unavailable, the loss of
38 revenue or business opportunities and the loss of trust from customers and
39 partners.

40 Many DDoS mitigation solutions have been proposed to protect organi-
41 zations' networks, servers and services. The traditional approach consists in
42 deploying specialized hardware security appliances that are fixed in terms of
43 strength, functionality and capacity. This means in particular that the loca-
44 tion and capacity (in terms of the volume of malicious traffic it can process)
45 of the defense appliances are determined in advance, before the DDoS attacks
46 actually take place. As explained e.g. by Fayaz et al. (2015), companies are
47 thus forced to over provision by deploying appliances capable of handling a
48 high but predefined volume of attack at several points in the network. A
49 second approach consists in using an external cloud-based DDoS protection
50 service. In this case, when under attack, all the incoming traffic to the tar-
51 geted service is diverted towards a cloud scrubbing center managed by a third
52 party. In the scrubbing center, the traffic is inspected and only the legitimate
53 traffic is routed back towards its destination. These cloud-based services are
54 more flexible and scalable than dedicated hardware appliances. They how-
55 ever raise concerns relative to customers' privacy violation and often lead to
56 increased latency (Alharbi and Aljuhani, 2017).

57 Network Function Virtualization (NFV) is a recent network architecture
58 concept in which network functions (e.g. network address translation, fire-
59 walling, domain name service, etc.) are implemented as software and de-
60 ployed as virtual machines running on general purpose commodity hardware
61 (Jakaria et al., 2016). Virtualization increases manageability, reliability and
62 performance of the network and allows a flexible and dynamic implemen-
63 tation of the network services, which significantly reduces the cost of the
64 infrastructure and simplifies the deployment of new services. These numer-
65 ous benefits have convinced operators to largely embrace virtualization of
66 network functions: see e.g. Donovan (2014) and Savi (2018).

67 NFV offers new possibilities to counter DDoS attacks. In particular, its

68 flexibility and reactivity allows to postpone the DDoS defense deployment
69 after the attack is detected. This allows to place adapted defense mechanisms
70 where they are needed and to launch them depending on the scale of the
71 attack (Fayaz et al., 2015). Moreover, NFV-based mitigation approaches do
72 not require the use of an external service provider, which reduces the privacy
73 and latency issues encountered by cloud-based DDoS mitigation.

74 As mentioned e.g in Alharbi and Aljuhani (2017), Silva et al. (2020)
75 and Jakaria et al. (2016), NFV is a promising technology to mitigate DDoS
76 attacks. However, in order to fully leverage its potential, some difficulties
77 should be overcome. First, virtual network functions (VNFs) are instan-
78 tiated on virtual machines. These virtual machines consume the limited
79 computing resources (CPU, memory,...) of the servers on which they run.
80 When designing an NFV-based infrastructure to counter an on-going DDoS
81 attack in a network, these limitations in the available computing resources
82 should be taken into account. The number of VNFs which can be instan-
83 tiated at each node of the network depends on the resources of the servers
84 located at this node. Second, each VNF has a limited filtering capacity and
85 can thus remove only part of the attack flow. The filtering capacity of a VNF
86 corresponds to the maximum amount of malicious flow an instance of this
87 VNF can stop. If the malicious flow going through a VNF is larger than its
88 filtering capacity, the excess malicious flow is forwarded in the network and
89 may thus reach its target. This translates into the fact that, in order to stop
90 all the malicious traffic of an attack, several VNFs may have to be placed at
91 different nodes on the paths used to route the flow between its source and its
92 target. A carefully optimized VNF placement strategy taking into account
93 both the limited computing resources in the network and the limited filtering
94 capacity of a VNF is thus needed.

95 In the present work, we focus on the deployment of an architecture based
96 on the NFV technology to secure a network against DDoS attacks. We
97 assume that the on-going attack has been detected and that its ingress points,
98 its volume and its target have been identified. Based on this information,
99 we seek to determine the optimal number and location of VNFs in order to
100 remove all the illegitimate traffic while trying to minimize the total cost of
101 the activated VNFs.

102 We take here the perspective of an internet service provider (ISP) aim-
103 ing at providing a DDoS mitigation service to its customers in a 5G net-
104 work. Among the key features of 5G networks is network slicing: see e.g.
105 Vyakaranam and Krishna (2018). Network slicing is an architecture in which

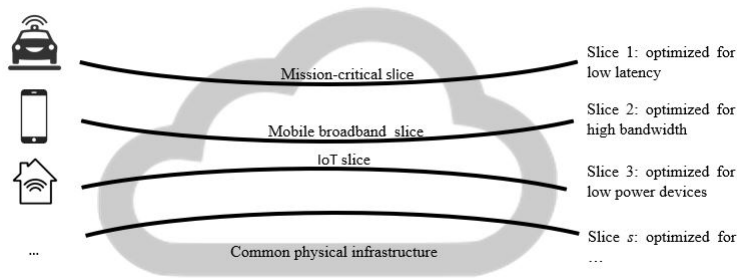


Figure 1: 5G network slicing

106 the physical network infrastructure managed by an ISP is partitioned into
 107 multiple virtual independent networks termed slices. Each slice is an iso-
 108 lated end-to-end network which is lent by the ISP to a single customer and
 109 is adapted to meet the specific requirements of this customer in terms of
 110 quality of service (bandwidth, reliability, latency, etc.). See Figure 1 for a
 111 graphical illustration of 5G network slicing. Network slicing thus provides
 112 an opportunity to the ISP to flexibly configure its physical network so as
 113 to simultaneously fulfill quality-of-service requirements that may strongly
 114 vary from one customer to the next. However, on each slice of the network,
 115 the routing of the flow will not be managed anymore by the ISP but by its
 116 customer which will rely on its own proprietary routing algorithms. This
 117 significantly enhances the difficulty for the ISP of providing a DDoS mitiga-
 118 tion service as it will not control the exact routing of the malicious flow that
 119 needs to be stopped.

120 Our main contributions are thus threefold. First, we present a robust op-
 121 timization (RO) model to optimally design an NFV-based DDoS mitigation
 122 infrastructure in the context of 5G network slicing. This model explicitly
 123 takes into account the fact that the ISP is not aware of the exact routing
 124 of the attack flow. This is done by considering the malicious flow routing
 125 as an input parameter of the optimization problem which is subject to un-
 126 certainty. To the best of our knowledge, this is the first time such a robust
 127 optimization model is investigated to design a DDoS mitigation infrastruc-
 128 ture in 5G networks. Second, we propose an efficient algorithm to solve
 129 the robust optimization problem. This algorithm relies on an adversarial
 130 approach which decomposes the problem into a master problem and an ad-
 131 versarial sub-problem. The master problem seeks to optimally place the

132 filtering VNFs while taking into account a limited number of possible mali-
133 cious flow routings. The adversarial sub-problem aims at finding the worst
134 flow routing for a given VNF infrastructure and is used to generate new rout-
135 ings, i.e. new constraints, to be taken into account in the master problem.
136 Moreover, as the adversarial sub-problem involves an exponential number
137 of decision variables, we develop a Branch & Price algorithm to solve it in
138 a computationally efficient way. Third, we provide the results of computa-
139 tional experiments carried out on medium-size randomly generated instances.
140 These results show that the proposed solution algorithm is able to efficiently
141 provide optimal or near-optimal solutions within short computation times.

142 The paper is organized as follows. We first review the related literature
143 in Section 2. We then provide in Section 3 a formal description of the prob-
144 lem, discuss its modeling as a robust optimization problem and present a
145 complexity analysis. We describe in Section 4 the adversarial solution ap-
146 proach proposed to solve this RO problem. Numerical results carried out on
147 medium-size randomly generated instances are provided in Section 5. Finally,
148 Section 6 gives a conclusion and some research perspectives.

149 2. Related works

150 We provide in this section a brief overview of the works closely related to
151 ours. We first discuss papers proposing NFV-based infrastructures for DDoS
152 mitigation. We then consider papers dealing with the optimal placement
153 of virtual network functions in a network for generic cases and focus on
154 two recent works studying the optimal placement of VNFs in a network for
155 the specific case of DDoS mitigation. Finally, we review the literature on
156 the network flow interdiction problem as this problem shares some common
157 features with our problem.

158 2.1. *NFV-based infrastructures for DDoS mitigation*

159 NFV-based infrastructures to counter DDoS attacks are investigated in
160 several recent papers. Fung and McCormick (2015) propose a solution based
161 on request prioritization to protect an online application server from a DDoS
162 attack. The incoming requests to the servers are categorized into two pri-
163 ority levels: requests from trusted sources are assigned a high priority and
164 are guaranteed to be served whereas requests from untrusted sources are as-
165 signed a low priority and will be served based on the resource availability
166 on the server. The proposed architecture makes use of a VNF for priority

167 assignment and flow dispatching. Another widely used mitigation strategy
168 against DDoS attacks is flow filtering: see e.g. Silva et al. (2020). Basically,
169 flow filtering consists in analyzing the information contained in the headers
170 of the data packets to block the malicious flow. The filtering process thus
171 exploits information such as the source and destination IP addresses, the
172 origin and destination ports or the network layer protocol to identify mali-
173 cious packets and drop them. Jakaria et al. (2016), Rashidi et al. (2018) and
174 Jakaria et al. (2019) investigate a DDoS mitigation framework in which this
175 filtering process is carried out by VNFs which are dynamically allocated as
176 needed depending on the volume of the attack. More precisely, their frame-
177 work aims at protecting an online product server against a specific type of
178 DDoS attacks, termed SYN floods, which exploit some weak points of the
179 TCP internet protocol. This framework involves a dispatcher/load balancer
180 which receives the incoming packets from the internet and distributes them
181 to filtering VNFs instantiated on commodity servers. These VNFs verify
182 the source IP address of each packet, drop the packet in case it is illegiti-
183 mate or forward it to the product server in case its source is white-listed.
184 Finally, Fayaz et al. (2015) and Alharbi and Aljuhani (2017) propose DDoS
185 mitigation infrastructures in which VNFs may have a variety of functions
186 depending on the type of the attack.

187 *2.2. Optimal placement of virtual network functions*

188 In their survey on network function placement, Li and Qian (2016) distin-
189 guish between two types of placement problems. The first one corresponds
190 to the case where independent network functions, i.e. functions which do
191 not interact with one another, should be placed in the network. The second
192 one, called service chaining, applies when each flow must traverse a prede-
193 fined sequence of network functions (such as firewall \rightarrow intrusion detection
194 system \rightarrow proxy) between its ingress point and its destination point in the
195 network. Note that the problem under study in this work belongs to the
196 first type of problem as we consider a single type of network functions. We
197 refer the reader to Demirci and Sagiroglu (2019) for a general overview of the
198 literature on the optimal placement of virtual network functions and focus
199 in what follows on the specific context of DDoS mitigation.

200 To the best of our knowledge, there are only two works dealing with
201 the problem of optimally placing VNFs in a network to counter an on-going
202 DDoS attack. Fayaz et al. (2015) develop the Bohatei system based on NFV
203 and SDN (software-defined networking). Their system includes a resource

204 manager which determines the type, number and location of VNFs to be
205 instantiated based on the available information on the ingress points, tar-
206 get, type and volume of the on-going attack so as to minimize the costs
207 related to the malicious flow traffic. They consider a case in which the mit-
208 igation of each type of DDoS attack (e.g. SYN flood, DNS amplification
209 or UDP flood) is a multi-step process requiring the use of different types
210 of network functions. They formulate the underlying optimization problem
211 as a mixed-integer linear program and solve it using a two-step heuristic.
212 Jakaria et al. (2019) consider an architecture involving two types of VNFs,
213 namely dispatchers and filtering agents, to counter SYN flood attacks. They
214 deploy these VNFs through virtual machines running on commodity servers.
215 The objective is to process all the incoming traffic while using a minimum
216 number of commodity servers. Their mathematical model is formulated as
217 a constraint satisfaction (SAT) problem (Apt, 2003). It takes into account
218 the limited computing resources of each commodity server, the limited band-
219 width of the links between the dispatchers and the filtering agents and the
220 relation between the packet filtering rate of a VNF and the computing re-
221 sources allocated to the virtual machine on which it is instantiated. Note
222 that, contrary to the problem under study here, both Fayaz et al. (2015) and
223 Jakaria et al. (2019) assume in their problem modeling that the flow of the
224 attack, once detected, can be flexibly routed towards the launched virtual
225 machines.

226 *2.3. Network flow interdiction problem*

227 In the network flow interdiction problem, an attacker and a defender take
228 measurements on a capacitated network. The defender seeks to maximize
229 the flow through the network, while the attacker suppresses some arcs to
230 minimize the maximum flow. Each arc has a removal cost. Thus, the goal
231 for the attacker is to select a subset of arcs to remove without exceeding
232 a fixed budget. The network interdiction problem is known to be an NP-
233 complete problem: see Phillips (1993) and Wood (1993). However, it can
234 be solved in polynomial time for certain categories of graphs such as planar
235 graphs (Phillips, 1993; Wollmer, 1964).

236 Different classes of the network flow interdiction problem are studied in
237 the literature: see e.g. Church et al. (2004). Baffier et al. (2018) investi-
238 gate an adaptive network interdiction flow problem. The defender aims to
239 maximize the flow value and the attacker seeks to minimize the remaining
240 flow value by removing a set of k links. The goal is to find a robust flow

241 against any k edge attack. A bilevel optimization framework is developed
242 to address this problem. Naoum-Sawaya and Ghaddar (2017) also formu-
243 late the problem as a bi-level mixed-integer program. An iterative cutting
244 plane algorithm is proposed and implemented in a branch-and-cut approach.
245 Lim and Smith (2007) study problems with discrete and continuous inter-
246 dictions. They describe a linearized model to optimize the discrete network
247 interdiction problem and compare it to a penalty model. For the continuous
248 case, they describe an optimal partitioning algorithm as well as a heuristic
249 procedure to estimate the optimal value of the objective function. Altner
250 et al. (2010) propose two classes of polynomially separable valid inequalities
251 for the Maximum Flow Network Interdiction Problem. An approximation
252 factor-preserving reduction from a simpler interdiction problem is also devel-
253 oped. Lei et al. (2018) consider maximum flow interdiction problem under
254 interdiction-effect uncertainties. The problem is characterized as a Stack-
255 elberg game. They consider risk-neutral and risk-averse behaviors of the
256 two players. Five bi-level/tri-level programming models for different risk-
257 preference combinations are investigated. An application of the network
258 interdiction problem to security issues is studied by Guo et al. (2016). The
259 problem is to optimally interdict illegal network flow in the context of the
260 containment of the flow of drugs through the US-Mexico border patrol. A
261 Stackelberg game model for network interdiction flow with a single source-
262 destination flow is presented. The proposed solution approach is based on
263 column generation and constraint generation algorithm. Fu and Modiano
264 (2019) propose a new paradigm for network interdiction that models sce-
265 narios. The interdiction is performed through injecting bounded-value flows
266 to maximally reduce the throughput of the residual network. They study
267 two problems under the paradigm: deterministic flow interdiction and ro-
268 bust flow interdiction. An algorithm with logarithmic approximation ratio is
269 developed.

270 Note that the present work investigates a defender-attacker problem
271 which significantly differs from the network flow interdiction problem. Namely,
272 in our case, the defender, i.e. the internet service provider, allocates secu-
273 rity resources without changing the network topology. Virtual functions are
274 deployed on network nodes in order to suppress attacking flows but this
275 security mechanism does not remove any component (node or link) in the
276 network. Our goal is to minimize the costs of VNFs deployment while grad-
277 ually eliminating the malicious flows, rather than destroying links to prevent
278 the attacker from reaching its target. This implies significant differences in

279 the mathematical formulations of the problem. The methodologies proposed
280 in the existing works can therefore not be directly applied to our problem.

281 3. Problem description and mathematical modeling

282 In this section, we first describe in a more formal way the optimization
283 problem under study and present the proposed robust optimization model.
284 We then provide a small illustrative example. Finally, we discuss the use
285 of aggregated filtering constraints in the problem formulation and study its
286 complexity status.

287 3.1. Problem definition

288 The network topology is modeled by a digraph $\mathcal{G} = (\mathcal{N}, \mathcal{L})$ in which \mathcal{N} ,
289 the set of nodes, represents specific equipment in the network and \mathcal{L} , the set
290 of arcs, corresponds to the links that can be used to route the traffic. The
291 routing of the traffic in the network is limited by the bandwidth b_l of each link
292 l . In practice, part of this bandwidth is used to route the legitimate traffic in
293 the network. In the present work, for the sake of simplicity, we assume that
294 the bandwidth consumed by the legitimate traffic is negligible as compared
295 to the one consumed by the illegitimate traffic. We thus consider that the
296 illegitimate traffic may use all the bandwidth of a link if needed.

297 The illegitimate traffic corresponding to the on-going DDoS attack is rep-
298 resented as a set \mathcal{A} of attacks: attack $a \in \mathcal{A}$ corresponds to an illegitimate
299 traffic of F^a Mbps between a source $s^a \in \mathcal{N}$ and the target $t \in \mathcal{N}$ of the
300 DDoS attack. Source nodes, $\{s_a, a \in \mathcal{A}\}$, are network access nodes (also
301 termed gateways) managed by the ISP. They are able to compute the num-
302 ber of incoming packets and to detect suspicious traffic entering the network.
303 In contrast, the target node t is a strategic node belonging to an external net-
304 work managed by a customer which subscribed to a security service provided
305 by the ISP. The ISP must thus secure this node against the on-going DDoS
306 attack but it is not allowed to install any software (i.e. to deploy VNFs) on
307 this node. The malicious traffic corresponding to the attack thus has to be
308 stopped before it reaches t .

309 As explained in the introduction, in the present work, we consider the
310 case in which an ISP lends slices of its physical network infrastructure to its
311 customers and each of these customers uses its own flow routing algorithms
312 to route the flow on the slice assigned to it. The result is that by the time the
313 ISP has to decide on the NFV-based DDoS mitigation infrastructure, it does

314 not know the exact routing of the malicious flow to be stopped. Let \mathcal{P}^a be
 315 the set of all potential paths between s^a and t for attack a . $\mathcal{N}^{a,p}$ (resp. $\mathcal{L}^{a,p}$)
 316 denotes the set of nodes (resp. the set of links) belonging to path $p \in \mathcal{P}^a$ and
 317 $\mathcal{P}^a(n)$ denotes the subset of paths of \mathcal{P}^a going through node n . The amount
 318 of malicious flow of attack $a \in \mathcal{A}$ on path $p \in \mathcal{P}^a$, denoted by $\tilde{f}^{a,p}$, is thus
 319 subject to uncertainty. However, even if the exact value of parameter $\tilde{f}^{a,p}$
 320 is unknown, there are some restrictions on its potential value. Namely, we
 321 know that the total amount of malicious flow routed on the paths belonging
 322 to \mathcal{P}^a may not be greater than F^a , the amount of illegitimate traffic of
 323 attack a . Moreover, the malicious flow routing must comply with the limited
 324 bandwidth of each link. These two pieces of information should be exploited
 325 as best as possible to avoid using more network resources than necessary for
 326 the DDoS attack mitigation.

327 In the considered DDoS mitigation framework, VNFs are used to filter and
 328 stop the illegitimate traffic before it reaches its target. **A VNF instantiated**
 329 **on a node $n \in \mathcal{N}$ of the network can be seen as a software running on the**
 330 **server located at node n and filtering the flow going through n .** As explained
 331 **in Section 2, this filtering process mainly consists in selectively stopping**
 332 **unwanted traffic by exploiting the information contained in the header of**
 333 **each data packet. This information can be the source, destination, port or**
 334 **routing protocol of the data packet to be processed.**The filtering capacity of
 335 **a VNF corresponds to the number of packets it can receive and process per**
 336 **second: if the malicious flow the VNF has to handle is larger than its filtering**
 337 **capacity, the excess flow is forwarded in the network and may thus reach its**
 338 **target. This filtering capacity is linked to the amount of computing resources**
 339 **consumed by the VNF on the server where it is instantiated. Indeed, data**
 340 **packets arriving at the VNF are first extracted and stored in memory. They**
 341 **then undergo several processing cycles on the available CPUs in order to**
 342 **analyze their content. Thus, the number of CPUs allocated to the VNF**
 343 **strongly limits its packet processing rate. Moreover, widely used filtering**
 344 **rules consist in analyzing the destination of a set of packets and storing them**
 345 **in memory. If there are too many packets targeting the same destination at**
 346 **the same time, these packets are considered as suspicious and are discarded.**
 347 **Consequently, the filtering process requires some memory to implement the**
 348 **malicious traffic filtering rules.** The set of available VNF types is described by
 349 $\mathcal{V} = \{1, \dots, V\}$. A VNF of type v is characterized by its filtering capacity ϕ^v ,
 350 its cost K^v and its computing resources consumption. The set of computing
 351 resources (CPU, memory, etc.) is denoted by $\mathcal{R} = \{1, \dots, R\}$. Let k^{rv} be the

352 amount of computing resource r required by the instantiation of one VNF of
 353 type v and Cap_n^r the amount of computing resource r available at node n .

354 **Table 1 summarizes the notation used to describe the input parameters**
 355 **of the various mathematical models throughout the paper.**

356 The optimization problem consists in identifying the location and number
 357 of VNFs to be placed in the network so as to stop all the malicious flow before
 358 it reaches its target, and this whatever its routing through the network,
 359 while minimizing the cost of the instantiated VNFs and complying with the
 360 limitations on the computing resources.

361 3.2. Mathematical formulation

362 We propose to handle this optimization problem using a robust opti-
 363 mization (RO) approach. A robust optimization problem is an optimization
 364 problem in which some parameters are subject to uncertainty. In a RO prob-
 365 lem, the uncertainty on the input parameters is not described in terms of
 366 probability distributions but rather by means of an uncertainty set contain-
 367 ing all the possible values that these parameters may take. Solving a RO
 368 problem consists in finding a solution which is feasible for any realization of
 369 the uncertain parameters in the uncertainty set and which provides the best
 370 possible value of the objective function. The reader is referred to Gorissen
 371 et al. (2015) for a practical introduction on robust optimization.

372 In the present case, the routing of the malicious flow in the network is
 373 not known by the ISP. The amount of malicious flow of attack $a \in \mathcal{A}$ on
 374 path $p \in \mathcal{P}^a$, $\tilde{f}^{a,p}$, can thus be seen as an uncertain input parameter for the
 375 problem of optimally placing VNFs to counter the DDoS attack. However,
 376 as mentioned in Subsection 3.1, even if the exact value of parameter $\tilde{f}^{a,p}$
 377 is unknown, its value should comply with two restrictions. First, for each
 378 attack a , the total flow routed in the network may not be larger than the
 379 total attack traffic, i.e. we have $\sum_{p \in \mathcal{P}^a} \tilde{f}^{a,p} \leq F^a$ for each attack $a \in \mathcal{A}$.
 380 Second, the flow routed on each link l of the network may not exceed the
 381 bandwidth b_l of this link. We thus have $\sum_{a \in \mathcal{A}} \sum_{p \in \mathcal{P}^a \text{ s.t. } l \in \mathcal{L}^{a,p}} \tilde{f}^{a,p} \leq b_l$ for
 382 each link l .

383 This means that the uncertain malicious flow routing, $\tilde{f} = \{\tilde{f}^{a,p} \text{ s.t. } a \in$
 384 $\mathcal{A}, p \in \mathcal{P}^a\}$, belongs to the uncertainty set \mathcal{U} defined by:

\mathcal{A}	Set of all on-going attacks to be stopped
\mathcal{C}	Collection of restricted sets of paths
\mathcal{G}	Graph representing the telecommunication network
\mathcal{L}	Set of links used to route the traffic in the network
$\mathcal{L}^{a,p}$	Set of all links belonging to path $p \in \mathcal{P}^a$
\mathcal{N}	Set of nodes, i.e. of pieces of equipment in the network
$\mathcal{N}^{a,p}$	Set of all nodes belonging to path $p \in \mathcal{P}^a$
$\mathcal{N}(\tilde{f})$	Subset of nodes through which part of the malicious flow transits when it is routed according to routing \tilde{f}
\mathcal{P}^a	Set of all paths between s^a and t
$\mathcal{P}^a(n)$	Subset of paths in \mathcal{P}^a such that $n \in \mathcal{N}^{a,p}$
\mathcal{P}_R^a	Restricted set of paths for attack a
\mathcal{R}	Set of computing resources
\mathcal{U}	Uncertainty set
\mathcal{U}_R	Restricted uncertainty set
\mathcal{V}	Set of available VNF types
A	Number of attacks
b_l	Bandwidth of link l
Cap_n^r	Amount of computing resource r available at node n
F^a	Total illegitimate traffic of attack a
$\tilde{f}^{a,p}$	Unknown amount of malicious flow of attack $a \in \mathcal{A}$ routed on path $p \in \mathcal{P}^a$
\tilde{f}	Unknown routing of the DDoS attack ; $\tilde{f} = \{\tilde{f}^{a,p} \text{ s.t. } a \in \mathcal{A}, p \in \mathcal{P}^a\}$
\bar{f}	Given routing belonging to the uncertainty set \mathcal{U}
K^v	Cost of instantiating a VNF of type v
k^{rv}	Amount of resource r required to instantiate a VNF of type v
R	Number of computing resources
s^a	Source, i.e. ingress point, of attack a
t	Target common to all on-going attacks
V	Number of available VNF types
\bar{x}	Given placement of the filtering VNFs in the network
ϕ^v	Filtering capacity of a VNF of type v

Table 1: Notation for the input parameters used in the various mathematical models

Problem RVNFD

x_n^v Number of VNFs of type v placed at node n

Problem TP

$d_n^{a,p}$ Amount of filtering capacity placed at node n allocated to stopping the malicious flow relative to attack a and routed on path p

Problem $DMP(\mathcal{U}_R)$

x_n^v Number of VNFs of type v placed at node n

Problems $AP(\bar{x})$, $RAP(\bar{x}, \mathcal{C})$ and $\underline{RAP}(\bar{x}, \mathcal{C})$

$f^{a,p}$ Amount of flow related to attack a routed on path p

z_n $z_n = 1$ if some malicious flow transits through n , to 0 otherwise

Table 2: Notation for the decision variables used in the various mathematical models

$$\mathcal{U} = \{ \tilde{f} \geq 0 \mid \sum_{p \in \mathcal{P}^a} \tilde{f}^{a,p} \leq F^a, \quad \forall a \in \mathcal{A} \\ \sum_{a \in \mathcal{A}} \sum_{p \in \mathcal{P}^a \text{ s.t. } l \in \mathcal{L}^{a,p}} \tilde{f}^{a,p} \leq b_l, \quad \forall l \in \mathcal{L} \}$$

385 Note that the first restriction on \tilde{f} is expressed as an inequality rather
386 than as an equality. Namely, in some cases, it may not be possible to route all
387 the malicious flow of the attack in the network due to the limited bandwidth
388 of the network links. In these cases, expressing the restriction as an equality
389 would lead to an empty uncertainty set. For the RO problem, this would
390 mean that there is no malicious flow routed in the network, i.e. no malicious
391 flow to be stopped by the VNF-based infrastructure, whereas in practice part
392 (but not all) of the attack flow will be routed in the network.

393 We introduce the integer decision variables x_n^v which represent the number
394 of VNFs of type v placed at node n : see Table 2 for a summary of the decision
395 variables used in the various mathematical models investigated in the paper.

396 Using the previously introduced notation, the robust virtual network
397 function deployment problem, which will be denoted by RVNFD in what
398 follows, is formulated as follows:

$$Z^* = \min \sum_{v \in \mathcal{V}} \sum_{n \in \mathcal{N}} K^v x_n^v \quad (1)$$

$$\sum_{v \in \mathcal{V}} k^{rv} x_n^v \leq \text{Cap}_n^r \quad \forall n \in \mathcal{N}, \forall r \in \mathcal{R} \quad (2)$$

$$\sum_{n \in \mathcal{N}(\tilde{f})} \sum_{v \in \mathcal{V}} \phi^v x_n^v \geq \sum_{a \in \mathcal{A}} \sum_{p \in \mathcal{P}^a} \tilde{f}^{a,p} \quad \forall \tilde{f} \in \mathcal{U} \quad (3)$$

$$x_t^v = 0 \quad \forall v \in \mathcal{V} \quad (4)$$

$$x_n^v \text{ integer} \quad \forall n \in \mathcal{N}, \forall v \in \mathcal{V} \quad (5)$$

399 The objective (1) is to minimize the total costs of the deployed VNFs.
400 Constraints (2) ensure that the VNFs installed at each node n do not con-
401 sume more than the available computing capacity for each computing re-
402 source. Constraints (3) translate the fact that we seek to avoid any damage
403 to the target by stopping all the malicious flow before it reaches it. In Con-
404 straints (3), $\mathcal{N}(\tilde{f}) = \{n \in \mathcal{N} \setminus \{t\} \mid \sum_{a \in \mathcal{A}} \sum_{p \in \mathcal{P}^a(n)} \tilde{f}^{a,p} > 0\}$ represents the
405 subset of nodes n through which part of the malicious flow transits when
406 considering the flow routing \tilde{f} . Constraints (3) impose that, for each possi-
407 ble routing \tilde{f} , the total filtering capacity installed on the nodes traversed
408 by a strictly positive amount of malicious flow in the routing \tilde{f} , i.e on the
409 nodes belonging to $\mathcal{N}(\tilde{f})$, is larger than the total malicious flow actually
410 routed through the network in \tilde{f} . Constraints (4) forbid any filtering at the
411 targeted node. Note that Constraints (3) are robust constraints that should
412 hold for any flow routing belonging to the uncertainty set \mathcal{U} .

413 3.3. *Small illustrative example*

414 Before discussing some theoretical aspects relative to the formulation and
415 complexity of Problem RVNFD, we provide a small illustrative example to
416 facilitate the understanding of the proposed models and methods.

417 Let us consider a small network \mathcal{G} including $|\mathcal{N}| = 5$ nodes and $|\mathcal{L}| =$
418 5 links, each one with a bandwidth of $b_l = 15\text{Mbps}$. The malicious flow
419 corresponding to the on-going DDoS attack enters the network at a gateway
420 located at node 1 and targets a critical customer node located at node 5:
421 we thus have $A = 1$, $s_1 = 1$ and $t = 5$. We consider $R = 1$ computing
422 resource corresponding to the number of CPUs available at each node: we
423 have $\text{Cap}_n^1 = 4$ CPUs available at node $n \in \{1, 2\}$ and $\text{Cap}_n^1 = 2$ CPUs

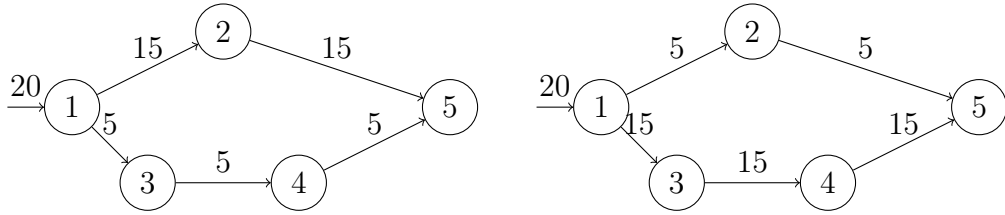


Figure 2: Small illustrative example: two possible routings for the malicious flow

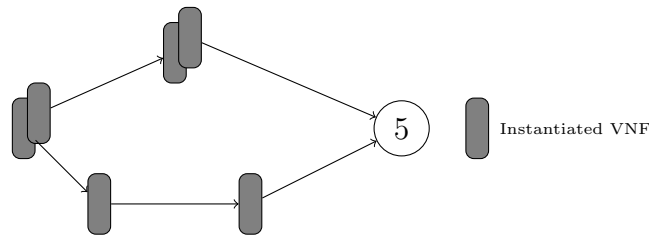


Figure 3: Small illustrative example: robust VNF placement

424 available at node $n \in \{3, 4\}$. There is a single type of filtering VNF, i.e.
 425 $V = 1$. Each instantiated VNF has a filtering capacity of $\phi_1 = 5\text{Mbps}$ and
 426 requires $k^{1,1} = 2$ CPUs.

427 Figure 2 displays two possible routings of the malicious flow. This one
 428 may use $|\mathcal{P}^1| = 2$ paths from node 1 to reach its target: path $p = 1$ corre-
 429 sponds to $1 \rightarrow 2 \rightarrow 5$ and path $p = 2$ to $1 \rightarrow 3 \rightarrow 4 \rightarrow 5$. The routing
 430 displayed on the left correspond to $\tilde{f}_{left} = (\tilde{f}_{left}^{1,1}, \tilde{f}_{left}^{1,2}) = (15, 5)$, the routing
 431 displayed on the right to $\tilde{f}_{right} = (\tilde{f}_{right}^{1,1}, \tilde{f}_{right}^{1,2}) = (5, 15)$. \tilde{f}_{left} and \tilde{f}_{right}
 432 are two elements (in fact two extreme points) of the uncertainty set \mathcal{U} .

433 By solving Problem RVNFD for this small instance, we obtain the VNF
 434 placement shown in Figure 3. It consists in placing two VNFs at nodes 1
 435 and 2 (i.e. $x_1^1 = x_2^1 = 2$) and one VNF at nodes 3 and 4 (i.e. $x_3^1 = x_4^1 = 1$.)
 436 Finally, Figure 4 presents how the malicious flow may be filtered by the
 437 instantiated VNFs in case it is routed according to \tilde{f}_{left} (see the network on
 438 the left) or according to \tilde{f}_{right} (see the network on the right). Note how, in
 439 both cases, all the malicious flow is filtered and stopped before it reaches the
 440 target located at node 5.

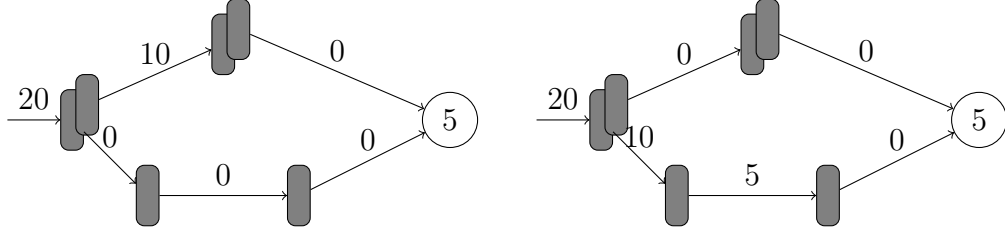


Figure 4: Small illustrative example: malicious flow filtering for two possible routings

3.4. Discussion on the aggregated attack filtering constraints

Constraints (3) can be seen as aggregated attack filtering constraints ensuring that the total filtering capacity installed on the set of nodes traversed by \bar{f} is larger than the total malicious flow routed through the network. As such, they do not guarantee that the filtering capacity installed on each potential path $p \in \mathcal{P}^a$ of each attack a is enough to stop all the flow related to attack a routed on this path, i.e. that the filtering capacity installed on each path $p \in \mathcal{P}^a$ is larger than $\bar{f}^{a,p}$. However, we show in what follows that, for any feasible solution \bar{x} of Problem **RVNFD** and any flow \bar{f} belonging to \mathcal{U} , we can find at least one allocation of the filtering capacity installed at each node n to the flows going through n such that all the malicious traffic can be filtered. This can be done by solving the following transportation problem denoted by **TP**.

Let $d_n^{a,p}$ be the decision variable representing the amount of filtering capacity installed at node n dedicated to stopping the malicious flow routed on path $p \in \mathcal{P}^a$, $a \in \mathcal{A}$.

$$Z_{TP}^* = \min \sum_{n \in \mathcal{N}} \sum_{a \in \mathcal{A}} \sum_{p \in \mathcal{P}^a(n)} d_n^{a,p} \quad (6)$$

$$\sum_{a \in \mathcal{A}} \sum_{p \in \mathcal{P}^a(n)} d_n^{a,p} \leq \sum_{v \in \mathcal{V}} \phi^v \bar{x}_n^v \quad \forall n \in \mathcal{N} \setminus \{t\} \quad (7)$$

$$\sum_{n \in \mathcal{N}^{a,p} \setminus \{t\}} d_n^{a,p} \geq \bar{f}^{a,p} \quad \forall a \in \mathcal{A}, \forall p \in \mathcal{P}^a \quad (8)$$

$$d_n^{a,p} \geq 0 \quad \forall n \in \mathcal{N}, \forall a \in \mathcal{A}, \forall p \in \mathcal{P}^a(n) \quad (9)$$

The objective (6) seeks to minimize the total amount of filtering capacity used to stop the malicious flow. Constraints (7) ensure that, at each node

459 n , the total amount of filtering capacity allocated to stop the flow routed on
 460 each path $p \in \mathcal{P}^a$ of each attack a going through node n is not larger than the
 461 amount of filtering capacity available at node n . Constraints (8) guarantee
 462 that, for each attack a and each path $p \in \mathcal{P}^a$ used to route the attack in \bar{f} ,
 463 the total amount of filtering capacity dedicated to p on the nodes belonging
 464 to it is large enough to stop all the malicious flow routed on p before it reaches
 465 t .

466 **Proposition 1.** *If \bar{x} is a feasible solution of Problem **RVNFD** and \bar{f} a flow*
 467 *belonging to the uncertainty \mathcal{U} , there exists at least one feasible solution for*
 468 *Problem **TP**, i.e. one allocation of the installed filtering capacity to the paths*
 469 *used by the attacks such that the total filtering capacity allocated to each path*
 470 *of each attack a is larger than $\bar{f}^{a,p}$.*

471 *Proof.* The proof is done by contradiction.

472 Let assume that Problem **TP** is unfeasible. It means that there exists
 473 a subset of attacks $\mathcal{A}' \subset \mathcal{A}$ and a subset of paths $\mathcal{P}'^a \subset \mathcal{P}^a$ for each at-
 474 tack $a \in \mathcal{A}'$ such that $\sum_{n \in \mathcal{N}'} \sum_{v \in \mathcal{V}} \phi^v \bar{x}_n^v < \sum_{a \in \mathcal{A}'} \sum_{p \in \mathcal{P}'^a} \bar{f}^{a,p}$ where $\mathcal{N}' =$
 475 $\cup_{a \in \mathcal{A}', p \in \mathcal{P}'^a} \mathcal{N}^{a,p}$. In other words, it exists a subset of paths $\mathcal{P}'^a, a \in \mathcal{A}'$, such
 476 that the total filtering capacity installed on the nodes belonging to \mathcal{N}' is
 477 insufficient to stop the flow going through these nodes.

478 Let us consider the routing f' defined by: $f'^{a,p} = \bar{f}^{a,p}$ if $a \in \mathcal{A}'$ and
 479 $p \in \mathcal{P}'^a$ and $f'^{a,p} = 0$ otherwise. We have $\mathcal{N}(f') = \mathcal{N}'$. As f' belongs to
 480 the uncertainty set \mathcal{U} and \bar{x} is a feasible solution of Problem **RVNFD**, the
 481 constraint $\sum_{n \in \mathcal{N}(f')} \sum_{v \in \mathcal{V}} \phi^v \bar{x}_n^v \geq \sum_{a \in \mathcal{A}'} \sum_{p \in \mathcal{P}'^a} f'^{a,p}$ should hold. This is in
 482 contradiction with the strict inequality written above.

483 □

484 In other words, solving Problem **TP** provides a VNF placement ensuring
 485 that all the malicious flow of the attack will be stopped provided we use an
 486 allocation of the filtering capacity to the paths actually used by the attack
 487 which complies with Constraints (7)-(9). Lemma 1 guarantees that such an
 488 allocation exists. However, solving Problem **RVNFD** does not guarantee that
 489 any allocation of the installed filtering capacity to the paths actually used
 490 by the attack will enable the ISP to block all the malicious flow.

491 3.5. Complexity analysis

492 **Proposition 2.** *Problem **RVNFD** is NP-hard, even if the uncertainty set \mathcal{U}*
 493 *contains a finite and discrete set of potential routings.*

494 *Proof.* The proof is done by reduction from the minimum set covering prob-
 495 lem.

496 Consider an instance I' of the minimum set covering problem. I' includes
 497 N potential location sites (indexed by $n = 1, \dots, N$) for the facilities and D
 498 demand points (indexed by $\delta_1, \dots, \delta_D$). For each demand point $\delta_d, d = 1, \dots, D$,
 499 we define the subset of potential location sites, $\mathcal{N}(\delta_d) \subset \{1, \dots, N\}$, which
 500 may cover it. The objective of the minimum set covering problem is to cover
 501 all demand points while minimizing the total number of opened facilities.

502 This instance of the minimum set covering problem can be transformed
 503 into an instance I of Problem RVNFD as follows. The graph $\mathcal{G} = (\mathcal{N}, \mathcal{L})$ has
 504 $N + 1$ nodes and N links. The nodes indexed by $n = 1 \dots N$ correspond to
 505 nodes where VNFs may be instantiated by the ISP and the node indexed by
 506 $N + 1$ corresponds to the target of the attack: we thus have $\mathcal{N} = \{1, \dots, N + 1\}$.
 507 There is a link $l \in \mathcal{L}$ between each node indexed by $n = 1 \dots N$ and the node
 508 indexed by $N + 1$. Each link has a bandwidth equal to $b_l = 1$. The DDoS
 509 attack enters the network at $A = N$ ingress points corresponding to the
 510 nodes indexed by $n = 1 \dots N$ (i.e. $s^a = a$ for $a = 1 \dots N$) and targets node
 511 $N + 1$ (i.e. $t = N + 1$). We set $F^a = \frac{1}{A}$ for each attack a .

512 Each attack a may thus use a single path to reach the target: for each a in
 513 \mathcal{A} , $|\mathcal{P}^a| = 1$ and the path indexed by $(a, 1)$ corresponds to $a \rightarrow t$. A routing
 514 in the network is thus a vector $f_d = (f_d^{1,1}, \dots, f_d^{a,1}, \dots, f_d^{A,1})$ describing the flow
 515 of malicious traffic on the single path of each attack. For each demand point
 516 $\delta_d, d = 1 \dots D$, of the minimum set covering problem, we add a routing f_d in
 517 the discrete uncertainty set \mathcal{U}_D with $f_d^{a,p} = \frac{1}{A}$ if node a belongs to $\mathcal{N}(\delta_d)$ and
 518 $f_d^{a,p} = 0$ otherwise.

519 We consider a single computing resource ($R = 1$) with $Cap_n^1 = 1$ for each
 520 node n in $\{1, \dots, N\}$. There is a single type of VNF indexed by $v = 1$ with a
 521 cost equal to $K^1 = 1$, a filtering capacity ϕ^1 equal to 1 and a consumption of
 522 the computing resource $k^{1,1}$ equal to 1. The total amount of malicious flow
 523 in any routing $f_d \in \mathcal{U}_D$, $\sum_{a \in \mathcal{A}} F^a$, is less than or equal to 1. Consequently,
 524 placing a VNF on any node belonging to $\mathcal{N}(f_d)$ suffices to ensure that the
 525 aggregated filtering constraints (3) will be satisfied for routing f_d .

526 Moreover, as the sets $\mathcal{N}(\delta_d)$ and $\mathcal{N}(f_d)$ coincide for each d , a demand
 527 point δ_d will be covered in instance I' as long as a VNF is instantiated on
 528 a node belonging to $\mathcal{N}(f_d)$ in instance I . As a consequence, determining,
 529 for instance I , the minimum cost VNF placement enabling to stop all the
 530 malicious flow, whatever its routing $f_d \in \mathcal{U}_D$, provides the minimum set of
 531 potential location sites covering all demand points in instance I' . Solving

532 instance I' of Problem RVNFD thus provides a solution to instance I of the
 533 minimum set covering problem.

534 As the minimum set covering problem is known to be NP-hard (see e.g.
 535 Korte and Vygen (2012)), the results follows.

536 □

537 Figure 5 illustrates this reduction on a small instance I' of the minimum
 538 set cover problem with $N = 3$ potential location sites (represented as dashed
 539 nodes indexed from 1 to 3) and $D = 4$ demand points represented by the
 540 nodes denoted by δ_1 to δ_4 . We have $\mathcal{N}(\delta_1) = \{1, 2\}$, $\mathcal{N}(\delta_2) = \{1, 2, 3\}$,
 541 $\mathcal{N}(\delta_3) = \{2\}$ and $\mathcal{N}(\delta_4) = \{3\}$. The corresponding graph is displayed at the
 542 top of Figure 5.

543 This instance of the minimum set covering problem can be transformed
 544 into an instance I of Problem RVNFD as follows. The corresponding graph
 545 $\mathcal{G} = (\mathcal{N}, \mathcal{L})$ has $N + 1 = 4$ nodes and $N = 3$ links: see the bottom part
 546 of Figure 5. The DDoS attack enters the network at $A = 3$ ingress points
 547 corresponding to the nodes indexed by $n = 1 \dots 3$ (i.e. $s^1 = 1$, $s^2 = 2$ and
 548 $s^3 = 3$) and targets node $t = 4$. We set $F^a = \frac{1}{A} = 0.33$ for each attack
 549 a . A routing $f_d = (f_d^{1,1}, f_d^{2,1}, f_d^{3,1})$ describes the amount of malicious flow
 550 routed on the single path $a \rightarrow 4$ that may be used by each attack a
 551 to reach the target. For each demand point $\delta_d, d = 1 \dots D$, of the minimum
 552 set covering problem, we add a routing f_d in the discrete uncertainty set
 553 \mathcal{U}_D such that $f_d^{a,p} = \frac{1}{A}$ if node a belongs to $\mathcal{N}(\delta_d)$ and $f_d^{a,p} = 0$ otherwise.
 554 This gives $f_1 = (0.33, 0.33, 0)$, $f_2 = (0.33, 0.33, 0.33)$, $f_3 = (0, 0.33, 0)$ and
 555 $f_4 = (0, 0, 0.33)$. We thus have $\mathcal{N}(\delta_d) = \mathcal{N}(f_d)$ for each $d = 1 \dots D$. We set
 556 $R = 1$, $Cap_n^1 = 1$ for $n = 1 \dots 3$, $V = 1$, $K^1 = 1$, $\phi_1 = 1$ and $k^{1,1} = 1$ as
 557 described in the proof of Proposition 2.

558 The optimal solution of instance I' consists in placing a VNF at nodes 2
 559 and 3 as this suffices to ensure that the aggregated filtering constraints (3)
 560 will be respected for all routings in the discrete uncertainty set \mathcal{U}_D . This
 561 gives an optimal solution of instance I which consists in opening a facility at
 562 the potential sites 2 and 3.

563 4. Solution approach

564 As explained e.g. by Gorissen et al. (2015), Problem RVNFD may seem
 565 intractable as such as the number of constraints (3) is infinite. Two main
 566 ways have been proposed in the literature to handle this difficulty.

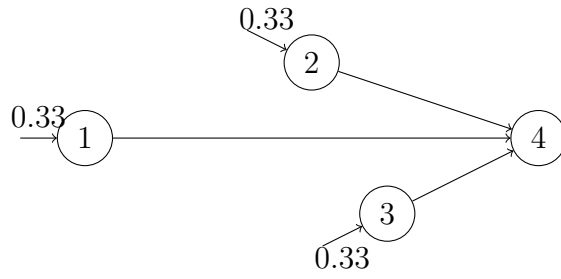
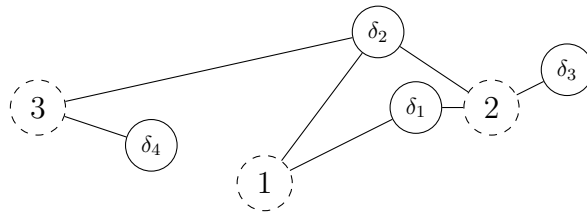


Figure 5: Reduction of an instance of the minimum set covering problem (top) into an instance of Problem RVNFD (bottom)

567 The first one consists in applying reformulation techniques which result
568 in the formulation of a deterministic problem with a finite number of con-
569 straints: see e.g. Bertsimas and Sim (2004). In our case, the use of these
570 reformulation techniques is not possible. Namely, the worst case reformation
571 of Constraints (3) would lead to the following expression:

$$\min_{\tilde{f} \in \mathcal{U}} \sum_{n \in \mathcal{N}} \mathbb{I} \left(\sum_{a \in \mathcal{A}} \sum_{p \in \mathcal{P}^a(n)} \tilde{f}^{a,p} > 0 \right) \sum_{v \in \mathcal{V}} \phi^v x_n^v - \sum_{a \in \mathcal{A}} \sum_{p \in \mathcal{P}^a} \tilde{f}^{a,p} > 0 \quad (10)$$

572 where $\mathbb{I} \left(\sum_{a \in \mathcal{A}} \sum_{p \in \mathcal{P}^a(n)} \tilde{f}^{a,p} > 0 \right)$ is an indicator function that is equal to
573 one if $\sum_{a \in \mathcal{A}} \sum_{p \in \mathcal{P}^a(n)} \tilde{f}^{a,p} > 0$ and zero otherwise. The resulting inner mini-
574 mization problem cannot be formulated as a linear program (but rather as a
575 mixed-integer linear program) due to the presence of this indicator function.
576 It is thus not possible to use the duality theory to reformulate it and obtain
577 a computationally tractable robust counterpart as is commonly done in this
578 type of reformulation approach.

579 The second possible way of solving a RO problem such as Problem **RVNFD**
580 consists in applying an adversarial approach. Such approaches are based on
581 the decomposition of the initial problem into a master problem and a sub-
582 problem. The master problem, called the decision maker problem in this
583 context, can be seen as a restricted version of the original RO problem in
584 which only a finite number of extreme points $\mathcal{U}_R \subset \mathcal{U}$ of the uncertainty
585 set (instead of the whole uncertainty set \mathcal{U}) are used to express the robust
586 constraints. This problem is a deterministic optimization problem with a
587 finite number of constraints and is thus computationally tractable. The sub-
588 problem is called the adversarial problem. Given the solution provided by
589 the decision maker problem, the adversarial problem seeks to find an extreme
590 point of \mathcal{U} for which this solution is infeasible. If no such extreme point can
591 be found, the current solution of the decision maker problem is optimal for
592 the initial RO problem. If such an extreme point is found, we add it to the
593 restricted set \mathcal{U}_R and reiterate the process. The finite convergence of this
594 algorithm is ensured by the fact that the uncertainty set \mathcal{U} has a finite num-
595 ber of extreme points. Adversarial approaches have been successfully used
596 to solve RO problems arising in a variety of applications: see among others
597 Bienstock and Özbay (2008), Attila et al. (2017), van Hulst et al. (2017) and
598 Agra et al. (2018).

599 *4.1. Adversarial approach*

600 The proposed adversarial approach thus iteratively solves the decision
 601 maker problem and the adversarial sub-problem. At each iteration, the deci-
 602 sion maker problem is solved using the current restricted uncertainty set \mathcal{U}_R
 603 and provides a placement of the VNFs \bar{x} which is optimal for this restricted
 604 uncertainty set. \bar{x} being given, the adversarial problem is solved to find the
 605 worst-case routing of the malicious flow for the VNF placement described by
 606 \bar{x} , i.e. to find an extreme point of \mathcal{U} which maximises the infeasibility of \bar{x} if it
 607 exists. In case such an extreme point is found, we update the restricted un-
 608 certainty set \mathcal{U}_R by adding the newly found routing \bar{f} and go on to the next
 609 iteration. Otherwise, \bar{x} is feasible for all extreme points of \mathcal{U} , the current
 610 VNF placement \bar{x} is optimal and the algorithm stops.

611 *4.1.1. Decision maker sub-problem*

612 The decision maker problem, denoted by $DMP(\mathcal{U}_R)$, can be formulated
 613 as follows:

$$Z_{DMP}^*(\mathcal{U}_R) = \min \sum_{v \in \mathcal{V}} \sum_{n \in \mathcal{N}} K^v x_n^v \quad (11)$$

$$\sum_{v \in \mathcal{V}} k^{rv} x_n^v \leq \text{Cap}_n^r \quad \forall n \in \mathcal{N}, \forall r \in \mathcal{R} \quad (12)$$

$$\sum_{n \in \mathcal{N}(\tilde{f})} \sum_{v \in \mathcal{V}} \phi^v x_n^v \geq \sum_{a \in \mathcal{A}} \sum_{p \in \mathcal{P}^a} \tilde{f}^{a,p} \quad \forall \tilde{f} \in \mathcal{U}_R \quad (13)$$

$$x_t^v = 0 \quad \forall v \in \mathcal{V} \quad (14)$$

$$x_n^v \text{ integer} \quad \forall n \in \mathcal{N}, \forall v \in \mathcal{V} \quad (15)$$

614 Problem $DMP(\mathcal{U}_R)$ thus displays the same structure as the initial RO
 615 problem but the number of Constraints (13) is now finite. Moreover, as will
 616 be shown by the numerical experiments provided in Section 5, in practice,
 617 the cardinality of \mathcal{U}_R , and as a consequence the number of Constraints (13)
 618 involved in the formulation, remain rather limited when implementing the
 619 adversarial approach. Problem $DMP(\mathcal{U}_R)$ can thus be directly solved by
 620 a mixed-integer linear programming solver with a reasonable computational
 621 effort.

622 4.1.2. Adversarial sub-problem

623 Let us now focus on the adversarial sub-problem. In order to formulate
624 it, we introduce the following decision variables:

- 625 - $f^{a,p}$: amount of malicious flow of attack a routed on path $p \in \mathcal{P}^a$,
- 626 - $z_n \in \{0, 1\}$: $z_n = 1$ if there is a positive amount of malicious flow transiting
627 through node n , 0 otherwise.

628 Given the current VNF placement \bar{x} , the maximum amount of malicious
629 flow which can reach its target can be found by solving the following mixed-
630 integer linear program, denoted by $AP(\bar{x})$.

$$Z_{AP}^*(\bar{x}) = \max \sum_{a \in \mathcal{A}} \sum_{p \in \mathcal{P}^a} f^{a,p} - \sum_{n \in \mathcal{N} \setminus \{t\}} \left(\sum_{v \in \mathcal{V}} \phi^v \bar{x}_n^v \right) z_n \quad (16)$$

$$\sum_{p \in \mathcal{P}^a} f^{a,p} \leq F^a \quad \forall a \in \mathcal{A} \quad (17)$$

$$\sum_{a \in \mathcal{A}} \sum_{p \in \mathcal{P}^a \text{ s.t. } l \in \mathcal{L}^{a,p}} f^{a,p} \leq b_l \quad \forall l \in \mathcal{L} \quad (18)$$

$$\sum_{a \in \mathcal{A}} \sum_{p \in \mathcal{P}^a(n)} f_p^a \leq \left(\sum_{a \in \mathcal{A}} F^a \right) z_n \quad \forall n \in \mathcal{N} \setminus \{t\} \quad (19)$$

$$f^{a,p} \geq 0 \quad \forall p \in \mathcal{P}^a \quad (20)$$

$$z_n \in \{0, 1\} \quad \forall n \in \mathcal{N} \quad (21)$$

631 The linear variables f thus describe the worst-case routing of the mali-
632 cious flow for the VNF placement \bar{x} . Constraints (17) ensure that, for each
633 attack, the total amount of flow of attack a routed through the network is
634 smaller than the total amount of flow of the attack F^a . Note that due to the
635 limited bandwidth of the network links, it might not be possible to route all
636 the flow of attack a through the network: Constraints (17) are thus formu-
637 lated as inequalities rather than as equalities. Constraints (18) guarantee
638 that the flow routed on each link does not exceed its bandwidth. In other
639 words, Constraints (17), (18) and (20) make sure that the solution of
640 problem $AP(\bar{x})$ provides a flow f belonging to the uncertainty set \mathcal{U} .

641 The objective function (16) seeks to maximize the amount of malicious
642 flow which will reach its target, i.e. which will not be filtered by a VNF
643 between its source and its target. Note that the filtering capacity $\sum_{v \in \mathcal{V}} \phi^v \bar{x}_n^v$
644 placed at node n can stop part of the malicious flow only if there is a
645 positive flow routed through node n , i.e. only if $\sum_{a \in \mathcal{A}} \sum_{p \in \mathcal{P}^a(n)} f_p^a > 0$.

646 $\sum_{a \in \mathcal{A}} \sum_{p \in \mathcal{P}^a} f_p^a - \sum_{n \in \mathcal{N} \setminus \{t\}} (\sum_{v \in \mathcal{V}} \phi^v \bar{x}_n^v) z_n$ thus computes the total amount
647 of unfiltered flow as the difference between the total flow routed through the
648 network, $\sum_{a \in \mathcal{A}} \sum_{p \in \mathcal{P}^a} f_p^a$, and the total amount of 'active' filtering capacity,
649 $\sum_{n \in \mathcal{N} \setminus \{t\}} (\sum_{v \in \mathcal{V}} \phi^v \bar{x}_n^v) z_n$. This 'active' filtering capacity is given by the sum
650 of the filtering capacities installed at the nodes n through which a positive
651 amount of malicious flow transits. Constraints (19) ensure that, for each
652 node n , variable z_n is equal to 1 as soon as there is some positive amount of
653 malicious flow which is routed through node n .

654 Note that, similar to what is done in Constraint (3) of the initial RO
655 problem, in the objective function (16) of the adversarial sub-problem, the
656 total amount of unfiltered flow is computed in an aggregate manner, i.e. by
657 looking at the total routed flow and at the total active filtering capacity on
658 all nodes of the network. In a feasible solution of problem $AP(\bar{x})$, this might
659 lead to an underestimation of the malicious flow which will reach its target.
660 Namely, we may have a subset of nodes \mathcal{N}' such that the total flow routed
661 through the nodes $n \in \mathcal{N}'$, $\sum_{a \in \mathcal{A}} \sum_{p \in \cup_{n \in \mathcal{N}'} \mathcal{P}^a(n)} f_p^a$, is smaller than the total
662 filtering capacity placed on these nodes, $\sum_{n \in \mathcal{N}'} (\sum_{v \in \mathcal{V}} \phi^v \bar{x}_n^v)$. In this case,
663 the actual filtering taking place at some of the nodes $n \in \mathcal{N}'$ is not equal to
664 $\sum_{v \in \mathcal{V}} \phi^v \bar{x}_n^v$ but to a smaller value. More precisely, the total filtering taking
665 place on the subset of nodes \mathcal{N}' is equal to $\sum_{a \in \mathcal{A}} \sum_{p \in \cup_{n \in \mathcal{N}'} \mathcal{P}^a(n)} f_p^a$ rather
666 than to $\sum_{n \in \mathcal{N}'} (\sum_{v \in \mathcal{V}} \phi^v \bar{x}_n^v)$. This means that the objective function (16)
667 overestimates the actual filtering taking place on the part of the network
668 corresponding to \mathcal{N}' and thus underestimates the amount of unfiltered ma-
669 licious flow. However, we show in what follows that such a situation cannot
670 occur in an optimal solution of $AP(\bar{x})$.

671 **Proposition 3.** *Any optimal solution of $AP(\bar{x})$ provides the worst-case rout-*
672 *ing for the given VNF placement \bar{x} .*

673 *Proof.* Let us consider a solution of $AP(\bar{x})$ in which there is at least one
674 subset of nodes \mathcal{N}' such that the total flow routed through the nodes $n \in \mathcal{N}'$,
675 $\sum_{a \in \mathcal{A}} \sum_{p \in \cup_{n \in \mathcal{N}'} \mathcal{P}^a(n)} f_p^a$, is smaller than $\sum_{n \in \mathcal{N}'} (\sum_{v \in \mathcal{V}} \phi^v \bar{x}_n^v)$. We show that
676 this solution cannot be optimal for $AP(\bar{x})$.

677 It is namely possible to build another feasible solution of $AP(\bar{x})$ by setting
678 to 0 the flow on all the paths belonging to $\cup_{n \in \mathcal{N}'} \mathcal{P}^a(n)$ and by setting z_n to
679 0 for all nodes $n \in \mathcal{N}'$. The objective value of the obtained solution will be
680 increased by $\sum_{n \in \mathcal{N}'} (\sum_{v \in \mathcal{V}} \phi^v \bar{x}_n^v) - \sum_{a \in \mathcal{A}} \sum_{p \in \cup_{n \in \mathcal{N}'} \mathcal{P}^a(n)} f_p^a > 0$, i.e. will be
681 strictly larger than the one of the initial solution. This latter can therefore

682 not be optimal. □

683 *4.2. Resolution of the adversarial sub-problem*

684 The adversarial sub-problem $AP(\bar{x})$ is a mixed-integer linear program
 685 which could theoretically be solved directly by a mathematical programming
 686 solver. However, the number of paths that could possibly be used to route
 687 the malicious flow of a given attack a between its source s^a and the target t ,
 688 and as a consequence the number of flow variables $f^{a,p}$, grows exponentially
 689 fast with the network size.

690 This difficulty may be overcome by using a column generation technique.
 691 In a column generation algorithm, we start solving problem $AP(\bar{x})$ with a
 692 restricted number of flow variables (i.e. of columns), which provides an initial
 693 feasible solution. This initial solution is then improved by iteratively adding
 694 new flow variables (i.e. by generating new columns) to the formulation of
 695 the problem until no more improving flow variables can be found.

696 Let $RAP(\bar{x}, \mathcal{C})$ be a restricted version of problem $AP(\bar{x})$ in which only
 697 a subset of the flow variables $f^{a,p}$ are explicitly considered. Here, \mathcal{C} denotes
 698 a collection of subsets of paths. More precisely, we have $\mathcal{C} = \{\mathcal{P}_R^a, a \in \mathcal{A}\}$
 699 where $\mathcal{P}_R^a \subset \mathcal{P}^a$ is the restricted subset of potential paths available for attack
 700 a taken into account in the problem formulation.

701 $RAP(\bar{x}, \mathcal{C})$ can be formulated as follows:

$$Z_{RAP}^*(\bar{x}, \mathcal{C}) = \max \sum_{a \in \mathcal{A}} \sum_{p \in \mathcal{P}_R^a} f_p^a - \sum_{n \in \mathcal{N} \setminus \{t\}} \left(\sum_{v \in \mathcal{V}} \phi^v \bar{x}_n^v \right) z_n \quad (22)$$

$$\sum_{p \in \mathcal{P}_R^a} f_p^a \leq F^a \quad \forall a \in \mathcal{A} \quad (23)$$

$$\sum_{a \in \mathcal{A}} \sum_{p \in \mathcal{P}_R^a \text{ s.t. } l \in \mathcal{L}^{a,p}} f_p^a \leq b_l \quad \forall l \in \mathcal{L} \quad (24)$$

$$\sum_{a \in \mathcal{A}} \sum_{p \in \mathcal{P}_R^a(n)} f_p^a \leq \left(\sum_{a \in \mathcal{A}} F^a \right) z_n \quad \forall n \in \mathcal{N} \setminus \{t\} \quad (25)$$

$$f_p^a \geq 0 \quad \forall p \in \mathcal{P}_R^a \quad (26)$$

$$z_n \in \{0, 1\} \quad \forall n \in \mathcal{N} \quad (27)$$

702 Note that $RAP(\bar{x}, \mathcal{C})$ displays the same structure as $AP(\bar{x})$ but the ob-
 703 jective and constraints are expressed using a limited number of flow variables

704 $f^{a,p}$, namely those corresponding to paths belonging to the restricted subset
 705 \mathcal{P}_R^a , for each attack $a \in \mathcal{A}$.

706 The column generation process, i.e. the process of adding new flow
 707 variables $f^{a,p}$, relies on the linear relaxation, denoted by $\underline{RAP}(\bar{x}, \mathcal{C})$, of
 708 $RAP(\bar{x}, \mathcal{C})$.

709 More precisely, at each iteration of the column generation algorithm, in
 710 order to identify improving flow variables to be added to the formulation, we
 711 first solve $\underline{RAP}(\bar{x}, \mathcal{C})$ with the current collection of path subsets \mathcal{C} . We then
 712 solve the pricing problem for each attack $a \in \mathcal{A}$. It consists in finding a flow
 713 variable $f^{a,p}$ with a positive reduced cost, i.e. a flow variable whose inclusion
 714 in the linear programming formulation might lead to an improvement of the
 715 objective function, or determining that no such variable exists. If at least
 716 one improving flow variable is found, we carry on with a new iteration of the
 717 algorithm. If no such variable is found, it means that the current solution
 718 of $\underline{RAP}(\bar{x}, \mathcal{C})$ is an optimal solution of the $\underline{AP}(\bar{x})$, the linear relaxation of
 719 $AP(\bar{x})$, and we stop.

720 Let α_a be the dual value of Constraint (23) relative to attack a , β_l the
 721 dual value of Constraint (24) relative to link l and γ_n the dual value of
 722 Constraint (25) relative to node n in the optimal solution of $\underline{RAP}(\bar{x}, \mathcal{C})$.
 723 The reduced cost of variable $f^{a,p}$ is given by $rc^{a,p} = 1 - (\alpha_a + \sum_{l \in \mathcal{L}^{a,p}} \beta_l +$
 724 $\sum_{n \in \mathcal{N}^{a,p}} \gamma_n)$.

725 Given an attack $a \in \mathcal{A}$, solving the pricing problem, i.e. identifying
 726 the variable $f^{a,p}$ with the largest reduced cost, thus amounts to finding the
 727 path $p \in \mathcal{P}^a$ with the smallest value of $\sum_{l \in \mathcal{L}^{a,p}} \beta_l + \sum_{n \in \mathcal{N}^{a,p}} \gamma_n$. This can
 728 be done by looking for the shortest path between s^a and t in the weighted
 729 digraph $(\mathcal{N}, \mathcal{L}, w)$ in which each link l has a weight of $w_l = \beta_l + \gamma_{dest(l)}$ where
 730 $dest(l)$ is the destination node of link l . This shortest path problem can be
 731 solved in polynomial time by Dijkstra's algorithm. If a variable $f^{a,p}$ with a
 732 positive reduced cost is found, the corresponding path is added to \mathcal{P}_R^a and
 733 the collection \mathcal{C} is updated accordingly.

734 Algorithm 1 provides a formal description of the column generation algo-
 735 rithm used to solve $\underline{AP}(\bar{x})$.

736 Note that Algorithm 1 solves to optimality the linear relaxation of $AP(\bar{x})$.
 737 In order to solve the original adversarial sub-problem $AP(\bar{x})$, which is a
 738 mixed-integer linear program, we consider two alternative ways of using it.

739 The first one corresponds to an exact Branch & Price algorithm. Basi-
 740 cally, a Branch & Price algorithm is a Branch & Bound method in which, at
 741 each node of the search tree, new variables may be added to the linear pro-

```

input : A VNF placement  $\bar{x}$  and a collection of path subsets  $\mathcal{C}$ 
output: An updated collection of path subsets  $\mathcal{C}$ 
begin
  repeat
    stop  $\leftarrow$  0
    solve  $\underline{RAP}(\bar{x}, \mathcal{C})$  with a linear programming solver
    get the dual values  $(\alpha, \beta, \gamma)$  of Constraints (23)-(25)
    for  $l=1$  to  $L$  do
      |  $w_l \leftarrow \beta_l + \gamma_{dest(l)}$ 
    end
    for  $a=1$  to  $A$  do
      | find the shortest path  $p_s$  between  $s^a$  and  $t$  in  $(\mathcal{N}, \mathcal{L}, w)$ 
      |  $rc_{p_s}^a \leftarrow 1 - (\alpha_a + \sum_{l \in \mathcal{L}_{p_s}^a} \beta_l + \sum_{n \in \mathcal{N}_{p_s}^a} \gamma_n)$ 
      | if  $rc_{p_s}^a > 0$  then
      | | stop  $\leftarrow$  1
      | |  $\mathcal{P}_R^a \leftarrow \mathcal{P}_R^a \cup \{p_s\}$ 
      | end
    end
  until stop = 0;
end

```

Algorithm 1: Column generation algorithm solving $\underline{AP}(\bar{x})$ to optimality

742 gramming relaxation. More precisely, the Branch & Price algorithm starts
 743 solving the restricted version of the adversarial sub-problem $RAP(\bar{x}, \mathcal{C})$ with
 744 an initial collection of path subsets \mathcal{C} , using a branch-and-bound method.
 745 At each node of the Branch & Bound search tree, we use Algorithm 1 to
 746 solve $\underline{AP}(\bar{x})$ and add new columns in the formulation (i.e. new paths in \mathcal{C}).
 747 When no new column can be generated by Algorithm 1, i.e. when the linear
 748 relaxation of the restricted master problem has been solved to optimality at
 749 the current Branch & Bound node, we either get an integer feasible solution
 750 of the initial problem $AP(\bar{x})$ or we branch on a fractional variable z_n to
 751 create new nodes in the search tree and continue with the Branch & Bound
 752 algorithm. The algorithm stops when there are no more open nodes in the
 753 search tree.

754 The second one is a heuristic algorithm. In this case, we first solve $\underline{AP}(\bar{x})$
 755 using Algorithm 1. When Algorithm 1 stops, we get the updated collection

756 of path subsets \mathcal{C} , reintroduce the integrality constraints on variables $z_n, n \in$
757 \mathcal{N} , and solve the restricted problem $RAP(\bar{x}, \mathcal{C})$ as a mixed-integer linear
758 program. Note that this algorithm may provide a sub-optimal solution of
759 $AP(\bar{x})$ as the collection of path subsets \mathcal{C} obtained by solving $\underline{AP}(\bar{x})$
760 not be the same as the one needed to obtain an optimal solution of $AP(\bar{x})$.

761 4.3. Summary of the proposed solution approach

762 The overall proposed solution approach is described by Algorithm 2 for
763 the case where the adversarial sub-problem is solved exactly and Algorithm 3
764 for the case where the adversarial sub-problem is solved heuristically.

765 In Algorithms 2 and 3, the restricted uncertainty set \mathcal{U}_R is initialized as
766 an empty set whereas the restricted path subset \mathcal{P}_R^a to be used for each attack
767 $a \in \mathcal{A}$ initially contains a single path, namely the shortest path in terms of
768 hops between the source of attack a and the target. Moreover, note that the
769 subsets $\mathcal{P}_R^a, a \in \mathcal{A}$, are not reinitialized at each iteration of the adversarial
770 algorithm. This means that the improving paths found while solving $AP(\bar{x}^i)$,
771 where \bar{x}^i denotes the solution of $DMP(\mathcal{U}_R)$ found at iteration i of the ad-
772 versarial algorithm, are part of the initial collection of path subsets provided
773 to Algorithm 1 when it will be used to solve $AP(\bar{x}^j)$, where \bar{x}^j denotes the
774 solution of $DMP(\mathcal{U}_R)$ found at any iteration $j > i$ of the adversarial algo-
775 rithm. Our preliminary numerical experiments namely showed that this was
776 more computationally efficient than reinitializing the subsets $\mathcal{P}_R^a, a \in \mathcal{A}$, at
777 each iteration of the adversarial algorithm.

778 5. Numerical results

779 5.1. Instances

780 We randomly generated a set of medium-size instances of the problem
781 following the indications provided by public data released by different cloud
782 and telecom providers.

783 **Network.** We used 4 internet network topologies. The first three ones
784 correspond to three internet networks described in the Internet Topology Zoo
785 library, IntelliFiber ($N = 73, L = 96$), Colt Telecom ($N = 153, L = 179$)
786 and Cogentco ($N = 197, L = 245$): see Knight et al. (2011) and Knight et al.
787 (2013) for more detail. We also used a topology corresponding to the former
788 network of the French company Free ($V = 120, E = 167$): see Ferre (2010).
789 Recall that the problem under study arises within the general context of
790 5G network slicing. As a consequence, we do not consider in our problem

```

begin
   $\mathcal{U}_R \leftarrow \emptyset$ 
  build the weighted digraph  $\mathcal{G} = (\mathcal{N}, \mathcal{L}, w)$  with  $w_l = 1, \forall l \in \mathcal{L}$ 
  for  $a=1$  to  $A$  do
    find the shortest path  $p_s$  between  $s^a$  and  $t$  in  $\mathcal{G}$ 
     $\mathcal{P}_R^a \leftarrow \{p_s\}$ 
  end
   $\mathcal{C} \leftarrow \{\mathcal{P}_R^a, a \in \mathcal{A}\}$ 
  repeat
    solve  $DMP(\mathcal{U}_R)$  and record the current VNF placement  $\bar{x}$ 
    solve  $RAP(\bar{x}, \mathcal{C})$  with a Branch & Price algorithm using
    Algorithm 1 to generate new columns at each node of the
    search tree and record the updated collection of path subsets
     $\mathcal{C}$ 
    if  $Z_{RAP}^*(\bar{x}, \mathcal{C}) > 0$  then
      record the optimal flow routing  $\bar{f}$ 
       $\mathcal{U}_R \leftarrow \mathcal{U}_R \cup \{\bar{f}\}$ 
    end
  until  $Z_{RAP}^*(\bar{x}, \mathcal{C}) \leq 0$ ;
end

```

Algorithm 2: Solution algorithm with an exact solution of the adversarial sub-problem

791 the whole physical network installed by the ISP but only the portion of this
792 network, i.e. the virtual network or slice, lent by the ISP to the customer
793 currently undergoing a DDoS attack. Thus, the bandwidth b_l of a link be-
794 tween two nodes does not correspond to the total bandwidth of the physical
795 link installed by the ISP between these nodes but only to the portion of
796 this bandwidth allocated to the virtual network dedicated to the customer
797 under attack. This is why we randomly generated values of b_l correspond-
798 ing to rather small transmission capacities. More precisely, the bandwidth
799 b_l of each link was randomly generated using a discrete distribution with a
800 support equal to $\{4.8, 12, 20, 40, 100\}$ Mbps.

801 **Computing resources.** $R = 2$ types of computing resources were taken
802 into account at each node: the number of CPUs and the memory. We con-
803 sidered three types of nodes: low computing capacity with $Cap = (8, 32)$,
804 medium computing capacity with $Cap = (40, 160)$ and high computing ca-

```

begin
   $\mathcal{U}_R \leftarrow \emptyset$ 
  build the weighted digraph  $\mathcal{G} = (\mathcal{N}, \mathcal{L}, w)$  with  $w_l = 1, \forall l \in \mathcal{L}$ 
  for  $a=1$  to  $A$  do
    find the shortest path  $p_s$  between  $s^a$  and  $t$  in  $\mathcal{G}$ 
     $\mathcal{P}_R^a \leftarrow \{p_s\}$ 
  end
   $\mathcal{C} \leftarrow \{\mathcal{P}_R^a, a \in \mathcal{A}\}$ 
  repeat
    solve  $DMP(\mathcal{U}_R)$  and record the current VNF placement  $\bar{x}$ 
    solve  $\underline{RAP}(\bar{x}, \mathcal{C})$  using Algorithm 1 and record the updated
      collection of path subsets  $\mathcal{C}$ 
    solve  $RAP(\bar{x}, \mathcal{C})$  as a mixed-integer linear program
    if  $Z_{RAP}^*(\bar{x}, \mathcal{C}) > 0$  then
      record the optimal flow routing  $\bar{f}$ 
       $\mathcal{U}_R \leftarrow \mathcal{U}_R \cup \{\bar{f}\}$ 
    end
  until  $Z_{RAP}^*(\bar{x}, \mathcal{C}) \leq 0$ ;
end

```

Algorithm 3: Solution algorithm with a heuristic solution of the adversarial sub-problem

805 capacity with $Cap = (400, 1600)$. In each considered network topology, we
 806 assign each node to a type according to its degree. Thus, nodes with a
 807 degree less than 2 were assigned a low computing capacity, nodes with a de-
 808 gree between 3 and 5 were assigned a medium computing capacity and nodes
 809 with a degree larger than 6 were assigned a high computing capacity. **Table 3**
 810 **provides a summary of the percentage of nodes assigned to each type (low,**
 811 **medium and high computing capacity) for each considered network topology.**

812 **VNFs.** $V = 1$ type of VNFs was considered requiring $\gamma^{1,1} = 4$ CPUs
 813 and $\gamma^{1,2} = 16$ units of memory, providing a filtering capacity of $\phi^n = 16$
 814 Mbps, with a unit cost of $K^1 = 130$.

815 **Attacks.** The number of sources was set to $A \in \{5, 10, 15, 20, 30, 40\}$. In
 816 each instance, the sources and target of the attack were randomly selected.
 817 The intensity F^a of each attack (in Mbps) was randomly generated following
 818 the normal distribution $\mathcal{N}(50, 25)$.

819 For each considered network topology and value of A , we randomly gen-

Topology	N	L	%Low	%Medium	%High
IntelliFiber	73	96	62%	37%	1%
Colt Telecom	153	179	72%	24%	4%
Cogentco	197	245	59%	39%	2%
Free	120	167	66%	28%	6%

Table 3: Percentage of nodes assigned to a low, medium or high computing capacity for each network topology

erated 5 instances, leading to a total of 140 instances.

5.2. Results

Each generated instance was solved using Algorithms 2 and 3. In both cases, the decision maker problem was solved as a mixed-integer linear program using the CPLEX 12.8.9 solver with the default settings. The adversarial sub-problem was solved using either the Branch & Price algorithm embedded in the SCIP 7.0.0 solver (Algorithm 2) or the simplex and Branch & Cut algorithms embedded in the CPLEX 12.8.9 solver (Algorithm 3). All tests were carried out on a PC running under Windows 10 equipped with an Intel Core i5-8350U processor (4 cores, frequency of 1.9GHz) and a 16 GB RAM with a 2400MHz speed. Note that the CPLEX 12.8.9 solver, in its default settings, is set to use a number of threads equal to the number of available cores whereas the SCIP 7.0.0 solver is by default single-threaded.

For each algorithm, each network topology and each considered value of A , we report in Table 4 the average value over the 5 corresponding instances of:

- *Cost*: the cost of the optimal VNF placement,
- *#IT*: the average number of iterations of the algorithm,
- *#P*: the total number of source-target paths added to \mathcal{C} by column generation over the course of the algorithm,
- *Time*: the total computation time in seconds of the algorithm.

Algorithm 3 is an approximate solution algorithm which may provide a solution which is not feasible for the initial robust optimization problem, i.e. for Problem RVNFD. Indeed, as the adversarial sub-problem is solved

844 heuristically, the amount of malicious flow that will reach the target may
845 be underestimated in some cases so that the filtering constraints (13) added
846 to the decision maker problem may not be tight enough. In order to es-
847 timate the impact of this heuristic resolution, we carry out the following
848 post-optimization analysis. We consider the optimal VNF placement \bar{x}_{app}
849 obtained with Algorithm 3. We solve problem $AP(\bar{x}_{app})$ exactly using the
850 Branch & Price algorithm. We then record AD the actual damage, i.e. the
851 amount of malicious flow which will actually reach its target, if we use the
852 VNF placement \bar{x}_{app} . We then compute the percentage of total unfiltered
853 flow $\%UF$ as $\%UF = \frac{100AD}{\sum_{a \in \mathcal{A}} F^a}$. We report in Table 4, for each set of 5
854 instances, $\#Inf$ the number of instances for which the solution obtained
855 with Algorithm 3 was infeasible and $Max\%UF$ the maximum percentage of
856 unfiltered flow.

857 Results from Table 4 first show that Algorithm 2 is able to provide optimal
858 solutions to the RO problem with a reasonable computational effort. Namely,
859 the average computation time, over the 140 considered instances, is 22s.
860 This performance is mainly explained by the fact that both the number
861 of iterations $\#IT$ of the algorithm (and as a consequence the number of
862 Constraints (13) of $DMP(\mathcal{U}_R)$) and the number of source-target paths $\#P$
863 generated by column generation (and as a consequence the number of flow
864 variables in $AP(\bar{x})$) stay limited.

865 However, the computation time of Algorithm 2 exceeds 60s for 10 out
866 of the 140 considered instances. This might be a problem as the decisions
867 on the VNFs deployment should be taken as quickly as possible after the
868 attack detection and identification. The approximate Algorithm 3 might
869 prove useful in such cases. It is namely able to provide optimal solutions
870 of the RO problem for 137 out of the 140 considered instances, and this
871 with an average computation time below 3s and a maximum computation
872 time of 25s. Moreover, for the 3 instances for which the solution provided
873 by Algorithm 3 did not comply with the original robust constraints (3), the
874 amount of malicious flow which could reach the target stays below 3%, which
875 seems acceptable.

876 6. Conclusion

877 This paper described a new robust optimization approach for the defense
878 against Distributed Denial of Service (DDoS) attacks in the context of 5G

Topology	Algorithm 2				Algorithm 3						
	<i>A</i>	<i>Cost</i>	<i>#IT</i>	<i>#P</i>	<i>Time</i>	<i>Cost</i>	<i>#IT</i>	<i>#P</i>	<i>Time(s)</i>	<i>#Inf</i>	<i>Max%UF</i>
IntelliFiber	5	936	9	25	3	936	10	25	1	0	0.00%
	10	1066	13	36	10	1066	11	34	1	0	0.00%
	15	1248	11	45	4	1248	6	36	1	0	0.00%
	20	988	9	35	11	988	8	35	0	0	0.00%
	30	1846	31	130	134	1846	17	107	2	0	0.00%
	40	1950	17	130	31	1950	15	109	1	0	0.00%
	Free	5	1092	12	27	10	1092	11	21	2	0
Colt	10	1326	10	41	4	1326	11	37	2	0	0.00%
	15	1378	5	60	3	1378	6	58	1	0	0.00%
	20	650	5	28	1	650	4	28	1	0	0.00%
	30	1092	8	71	3	1092	5	71	1	0	0.00%
	40	1352	7	63	4	1352	6	69	1	0	0.00%
	5	1118	17	42	12	1118	19	40	5	1	1,16%
	10	806	7	35	2	806	7	41	2	0	0.00%
Cogentco	15	1170	20	63	38	1170	19	59	5	0	0.00%
	20	1092	10	63	14	1092	11	59	3	0	0.00%
	30	754	7	71	4	754	4	70	2	0	0.00%
	40	1118	9	66	9	1118	6	40	2	0	0.00%
	5	546	13	58	9	546	14	45	5	1	2,88%
	10	754	14	70	24	754	14	50	6	0	0.00%
	15	1222	18	95	20	1222	14	73	5	0	0.00%
20	910	21	81	47	910	14	75	6	0	0.00%	
30	728	13	84	34	728	9	78	4	0	0.00%	
40	1066	20	101	107	1066	13	87	7	1	0,18%	

Table 4: Numerical results

879 network slicing. More precisely, we considered the problem of optimally de-
880 ploying virtual network functions in order to stop an ongoing DDOS attack.
881 We assumed that the target, sources and volume of the attack are identified
882 but that the exact routing of the illegitimate traffic on the network is not
883 known. To take into account these uncertainties, we proposed a robust opti-
884 mization (RO) model and developed an adversarial approach to solve it. This
885 iterative approach is based on the decomposition of the initial problem into
886 a master problem and a sub-problem. The master problem is a restricted
887 version of the original RO problem in which only a finite number of possible
888 malicious flow routings are used to express the robust constraints. Consid-
889 ering the current placement of VNF provided by the solution of the master
890 problem, the adversarial sub-problem seeks to find a malicious flow routing
891 that maximizes the amount of attack reaching its target. We tested the ef-
892 ficiency of our algorithms on medium-sized randomly generated instances.
893 The results of computation experiments show that our approach is able of
894 providing optimal solutions in short computation times.

895 Current work suggests several possible directions for future research. In
896 terms of problem solving, it might be possible to further improve the decom-
897 position approach by carrying out a polyhedral study of the problem and
898 developing new valid inequalities to help solving it more quickly. As for the
899 problem modeling, a first research direction could consist in studying a disag-
900 gregated formulation of the robust filter constraints. This could ensure that
901 the instantiated VNFs will be able to stop all the malicious flows regardless
902 of the allocation of filtering capacities. It would also be interesting to study
903 how the legitimate traffic, which will consume network resources and whose
904 routing is also unknown, could be taken into account in the model.

905 **Acknowledgement**

906 The authors would like to thank Kahina Lazri and Paul Chaignon (Orange
907 Labs Products & Services) for their highly appreciated help in understanding
908 and modeling the optimization problem. We are also grateful to Claudia
909 D'Ambrosio (Laboratoire d'Informatique de l'École Polytechnique, CNRS,
910 France) and Andrea Lodi (Ecole Polytechnique Montréal, Canada) for their
911 fruitful advice on the problem modeling and solving.

912 **References**

- 913 Agra, A., Christiansen, M., Hvattum, L.M., Rodrigues, F., 2018. Robust
914 optimization for a maritime inventory routing problem. *Transportation*
915 *Science* 52, 509–525.
- 916 Akpakwu, G.A., Silva, B.J., Hancke, G.P., Abu-Mahfouz, A.M., 2018. A
917 survey on 5G networks for the Internet of Things: Communication tech-
918 nologies and challenges. *IEEE Access* 6, 3619–3647.
- 919 Alharbi, T., Aljuhani, A., 2017. Holistic DDoS mitigation using NFV, in:
920 2017 IEEE 7th Annual Computing and Communication Workshop and
921 Conference CCWC.
- 922 Altner, D.S., Ergun, Ö., Uhan, N.A., 2010. The maximum flow network inter-
923 diction problem: valid inequalities, integrality gaps, and approximability.
924 *Operations Research Letters* 38, 33–38.
- 925 Apt, K.R., 2003. *Principles of Constraint Programming*. Cambridge Univer-
926 sity Press.
- 927 Attila, Ö.N., Agra, A., Akartunali, K., Arulselvan, A., 2017. A decomposi-
928 tion algorithm for robust lot sizing problem with remanufacturing option,
929 in: Gervasi, O., Murgante, B., Misra, S., Borruso, G., Torre, C.M., Rocha,
930 A.M.A., Taniar, D., Apduhan, B.O., Stankova, E., Cuzzocrea, A. (Eds.),
931 *Computational Science and Its Applications – ICCSA 2017*, Springer In-
932 ternational Publishing. pp. 684–695.
- 933 AWS, 2020. AWS Shield - Threat landscape report Q1 2020. [https://aws-
934 shield-tlr.s3.amazonaws.com/2020-Q1_AWS_Shield_TLR.pdf](https://aws-shield-tlr.s3.amazonaws.com/2020-Q1_AWS_Shield_TLR.pdf). Accessed
935 2020-12-19.
- 936 Baffier, J.F., Poirion, P.L., Suppakitpaisarn, V., 2018. Bilevel model for
937 adaptive network flow problem. *Electronic Notes in Discrete Mathematics*
938 64, 105–114. 8th International Network Optimization Conference - INOC
939 2017.
- 940 Berard, D., 2018. DDoS breach costs rise to over \$2M for enterprises
941 finds kaspersky lab report. [https://usa.kaspersky.com/about/press-
942 releases/2018_ddos-breach-costs-rise-to-over-2m-for-enterprises-finds-
943 kaspersky-lab-report](https://usa.kaspersky.com/about/press-releases/2018_ddos-breach-costs-rise-to-over-2m-for-enterprises-finds-kaspersky-lab-report). Accessed 2020-12-19.

- 944 Bertsimas, D., Sim, M., 2004. The price of robustness. *Operations Research*
945 52, 35–53.
- 946 Bienstock, D., Özbay, N., 2008. Computing robust basestock levels. *Discrete*
947 *Optimization* 5, 389 – 414.
- 948 Church, R.L., Scaparra, M.P., Middleton, R.S., 2004. Identifying critical in-
949 frastructure: the median and covering facility interdiction problems. *An-*
950 *nals of the Association of American Geographers* 94, 491–502.
- 951 Demirci, S., Sagiroglu, S., 2019. Optimal placement of virtual network func-
952 tions in software defined networks: A survey. *Journal of Network and*
953 *Computer Applications* 147, 102424.
- 954 Donovan, J., 2014. How SDN enabled innovations will impact AT&T’s plans
955 to transform it’s infrastructure. www.bit.ly/1RQFMko. Accessed 2020-10-
956 01.
- 957 Fayaz, S.K., Tobioka, Y., Sekar, V., Bailey, M., 2015. Bohatei: Flexible and
958 elastic DDoS defense, in: 24th USENIX Security Symposium (USENIX
959 Security 15), pp. 817–832.
- 960 FBI, 2020. Cyber actors exploiting built-in network proto-
961 cols to carry out larger, more destructive distributed de-
962 nial of service attacks. [https://dd80b675424c132b90b3-](https://dd80b675424c132b90b3-e48385e382d2e5d17821a5e1d8e4c86b.ssl.cf1.rackcdn.com/external/fbi-private-industry-notification-20200721-002.pdf)
963 [e48385e382d2e5d17821a5e1d8e4c86b.ssl.cf1.rackcdn.com/external/fbi-](https://dd80b675424c132b90b3-e48385e382d2e5d17821a5e1d8e4c86b.ssl.cf1.rackcdn.com/external/fbi-private-industry-notification-20200721-002.pdf)
964 [private-industry-notification-20200721-002.pdf](https://dd80b675424c132b90b3-e48385e382d2e5d17821a5e1d8e4c86b.ssl.cf1.rackcdn.com/external/fbi-private-industry-notification-20200721-002.pdf). Accessed 2020-12-19.
- 965 Ferre, L., 2010. Free SAS domestic network.
966 [https://fr.wikipedia.org/wiki/Free_\(entreprise\)](https://fr.wikipedia.org/wiki/Free_(entreprise)). Accessed 2020-12-19.
- 967 Fu, X., Modiano, E., 2019. Network interdiction using adversarial traffic
968 flows, in: *IEEE INFOCOM 2019-IEEE Conference on Computer Commu-*
969 *nications*, IEEE. pp. 1765–1773.
- 970 Fung, C.J., McCormick, B., 2015. Vguard: A distributed denial of ser-
971 vice attack mitigation method using network function virtualization, in:
972 2015 11th International Conference on Network and Service Management
973 (CNSM), pp. 64–70.

- 974 Fysarakis, K., Askoxylakis, I., Manifavas, C., Soultatos, O., Papaefstathiou,
975 I., Katos, V., 2016. Which IoT protocol? comparing standardized ap-
976 proaches over a common M2M application., in: 2016 IEEE Global Com-
977 munications Conference (Globecom).
- 978 Gorissen, B.L., Yanikoglu, I., den Hertog, D., 2015. A practical guide to
979 robust optimization. *Omega* 53, 24 – 137.
- 980 Grawe, K., 2020. Link11 h1 2020 DDoS report reveals a
981 resurgence in DDoS attacks during COVID-19 lockdowns.
982 [https://www.link11.com/en/blog/threat-landscape/h1-2020-link11-](https://www.link11.com/en/blog/threat-landscape/h1-2020-link11-ddos-report-en/)
983 [ddos-report-en/](https://www.link11.com/en/blog/threat-landscape/h1-2020-link11-ddos-report-en/). Accessed 2020-12-19.
- 984 Guo, Q., An, B., Zick, Y., Miao, C., 2016. Optimal interdiction of illegal
985 network flow .
- 986 van Hulst, D., den Hertog, D., Nuijten, W., 2017. Robust shift generation in
987 workforce planning. *Computational Management Science* 14, 115–134.
- 988 Jakaria, A.H.M., Rahman, M.A., Fung, C., 2019. A requirement-oriented de-
989 sign of NFV topology by formal synthesis. *IEEE Transactions on Network*
990 *and Service Management* 16, 1739–1753.
- 991 Jakaria, A.H.M., Yang, W., Rashidi, B., Fung, C., Rahman, M.A., 2016.
992 Vfence: A defense against distributed denial of service attacks using net-
993 work function virtualization, in: 2016 IEEE 40th Annual Computer Soft-
994 ware and Applications Conference (COMPSAC), pp. 431–436.
- 995 Khandelwal, S., 2016. 602 Gbps! this may have been the largest DDoS attack
996 in history. www.thehackernews.com/2016/01/biggest-ddos-attack.html.
997 Accessed 2020-12-19.
- 998 Knight, S., Nguyen, H.X., Falkner, N., Bowden, R., Roughan, M., 2011. The
999 internet topology zoo. *IEEE Journal on Selected Areas in Communications*
1000 29, 1765–1775. doi:10.1109/JSAC.2011.111002.
- 1001 Knight, S., Nguyen, H.X., Falkner, N., Bowden, R., Roughan, M., 2013. The
1002 internet topology zoo. <http://www.topology-zoo.org/index.html>. Accessed
1003 2020-12-19.

- 1004 Korte, B., Vygen, J., 2012. Combinatorial Optimization: Theory and Algo-
1005 rithms. Springer-Verlag.
- 1006 Lei, X., Shen, S., Song, Y., 2018. Stochastic maximum flow interdiction
1007 problems under heterogeneous risk preferences. Computers and Operations
1008 Research 90, 97–109.
- 1009 Li, X., Qian, C., 2016. A survey of network function placement, in: 2016
1010 13th IEEE Annual Consumer Communications Networking Conference
1011 (CCNC), pp. 948–953. doi:10.1109/CCNC.2016.7444915.
- 1012 Lim, C., Smith, J.C., 2007. Algorithms for discrete and continuous multicom-
1013 modity flow network interdiction problems. IIE Transactions 39, 15–26.
- 1014 Naoum-Sawaya, J., Ghaddar, B., 2017. Cutting plane approach for the max-
1015 imum flow interdiction problem. Journal of the Operational Research So-
1016 ciety 68, 1553–1569.
- 1017 Netscout Systems, 2020. Netscout threat intelligence report.
1018 [https://www.netscout.com/sites/default/files/2020-02/SECR_001_EN-](https://www.netscout.com/sites/default/files/2020-02/SECR_001_EN-2001_Web.pdf)
1019 [2001_Web.pdf](https://www.netscout.com/sites/default/files/2020-02/SECR_001_EN-2001_Web.pdf). Accessed 2020-12-19.
- 1020 Phillips, C.A., 1993. The network inhibition problem, in: Proceedings of the
1021 Twenty-Fifth Annual ACM Symposium on Theory of Computing, Associ-
1022 ation for Computing Machinery, New York, NY, USA. pp. 776–785.
- 1023 Rahimi, H., Zibaenejad, A., Safavi, A.A., 2018. A novel IoT architecture
1024 based on 5G-IoT and next generation technologies, in: 2018 IEEE 9th
1025 Annual Information Technology, Electronics and Mobile Communication
1026 Conference (IEMCON), pp. 81–88. doi:10.1109/IEMCON.2018.8614777.
- 1027 Rashidi, B., Fung, C., Rahman, M., 2018. A scalable and flexible DDoS
1028 mitigation system using network function virtualization, in: NOMS 2018
1029 - 2018 IEEE/IFIP Network Operations and Management Symposium, pp.
1030 1–6.
- 1031 Savi, J., 2018. With OPNFV, Orange plans a full-scale rollout of net-
1032 work functions virtualization. [https://thenewstack.io/orange-relies-opnfv-](https://thenewstack.io/orange-relies-opnfv-transform-networks-future/)
1033 [transform-networks-future/](https://thenewstack.io/orange-relies-opnfv-transform-networks-future/). Accessed 2020-12-19.

- 1034 Silva, F.S.D., Silva, E., Neto, E.P., Lemos, M., Neto, A.J.V., Esposito, F.,
1035 2020. A taxonomy of DDoS attack mitigation approaches featured by SDN
1036 technologies in IoT scenarios. *Sensors* 20, 3078.
- 1037 Vyakaranam, N., Krishna, D., 2018. 5G: Network as a service -
1038 how 5G enables the telecom operators to lease out their network.
1039 [https://netmanias.com/en/post/blog/13311/5g/5g-network-as-a-service-](https://netmanias.com/en/post/blog/13311/5g/5g-network-as-a-service-how-5g-enables-the-telecom-operators-to-lease-out-their-network)
1040 [how-5g-enables-the-telecom-operators-to-lease-out-their-network.](https://netmanias.com/en/post/blog/13311/5g/5g-network-as-a-service-how-5g-enables-the-telecom-operators-to-lease-out-their-network) Ac-
1041 cessed 2020-12-19.
- 1042 Wollmer, R., 1964. Removing arcs from a network. *Operations Research* 12,
1043 934–940.
- 1044 Wood, R.K., 1993. Deterministic network interdiction. *Mathematical and*
1045 *Computer Modelling* 17, 1–18.