Computer Networks 222 (2023) 109526

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Distributed dual-layer autonomous closed loops for self-protection of 5G/6G IoT networks from distributed denial of service attacks☆

Pablo Benlloch-Caballero *, Qi Wang, Jose M. Alcaraz Calero

University of the West of Scotland, School of Computing, Engineering and Physical Sciences, United Kingdom

ARTICLE INFO

Keywords:
Self-managed networks
Autonomous control loop
5G/6G networks
Multi-stakeholder distributed dual-layer self-protection
DDoS detection and mitigation system
Distributed network self-protection system

ABSTRACT

Internet of Things (IoT) is a major application area of the Fifth-Generation (5G) and beyond capable of providing massive machine-type communications (mMTC) at a large scale. It enables a wide range of applications such as smart cities, smart grids, smart factories and so on. In light of the huge number of devices involved, it is prohibitive to manage the massive large-scale cyber security scenarios manually. Therefore, closed automation loops are essential to automate such management. This paper proposes a new cognitive closed loop system to offer distributed dual-layer self-protection capabilities to battle against Distributed Denial of Service (DDoS) attacks. The proposed system features the novel usage of concurrent autonomous closed-loops for the different stakeholders' business roles: Digital Service Providers (DSPs) and Infrastructure Service Providers (ISPs) respectively, suitable to provide a multi-layer self-protection defence mechanisms across multiple administrative domains. It has been designed, implemented and experimentally validated. Empirical results have shown that there is a high potential in the collaboration between the stakeholders to achieve the common goal of self-protection of infrastructures. It makes a major difference in the performance of the whole infrastructure for detecting, analysing and mitigating the threat when the proposed distributed dual-layer loops are applied instead of a standalone loop. The system has achieved a 78.12% of effectiveness compared with a 4.73% of the standalone counterpart, for a large scale attack when stopping 256 infected devices. Also, the proposed system has achieved a response time of 18 s whereas the standalone has required 57 s, achieving an optimization of performance of 316%.

1. Introduction

We are currently witnessing an exponentially increasing use of Internet-of-Things (IoT) devices to date, and growth is expected to remain the same trend [1]. One of the open research questions to be addressed, according to Kaspersky [2], is whether to prioritize the protection of IoT devices or to protect the networks from IoT device attacks. This is due to the interest that wearables, Smart TVs and other gadgets have risen in cyber-criminals, who see an opportunity to use such IoT devices as bots or zombies in cyber attacks. These devices can be used to deploy Distributed Denial of Service (DDoS) attacks, creating botnets capable of leaving large companies and communities without service, and even countries. As reported in October 2017, more than 2 million IoT devices were infected by Reaper, an evolution of the old Mirai DDoS, and it is considered more virulent than its predecessor due to the ease of device infection [3]. The current 5G and the foreseen 6G multi-tenant networks are the most predominant networks to be used to scale up the communication networks that interconnect such IoT devices. Therefore, they are also considered susceptible targets of those attackers.

One of the main characteristics in the development of the new generation (5G and 6G) networks that has been promoted is the softwarization and virtualization of network services. In this way, different virtualized systems can be hosted on the same physical infrastructure, providing service to several companies, and sharing space and physical resources. This sharing capabilities is commonly refereed as multi-tenancy. The virtualization of network devices and the usage of tunnelling of network traffic, allows maintaining isolated traffic among the tenants that host the physical infrastructure. However, the usage of

☆ This work is funded in part by the European Commission under Grant Agreements H2020-SU-DS-2018-2019-2020/101020259 (ARCADIAN-IoT: Autonomous Trust, Security and Privacy Management Framework for IoT), H2020-ICT-2020-2/101017226 (6G BRAINS: Bringing Reinforcement learning Into Radio Light Network for Massive Connections) and HORIZON-JU-SNS-2022-STREAM-B-01-04/101095933 (RIGOUROUS: secuRe desIGn and deplOyment of trUsthwoRthy cOntinUum computing 6G Services).

* Corresponding author.
E-mail addresses: Pablo.Benlloch-Caballero@uws.ac.uk (P. Benlloch-Caballero), Qi.Wang@uws.ac.uk (Q. Wang), Jose.Alcaraz-Calero@uws.ac.uk (J.M. Alcaraz Calero).

https://doi.org/10.1016/j.comnet.2022.109526
Received 11 August 2022; Received in revised form 4 November 2022; Accepted 11 December 2022
Available online 21 December 2022

tunnelling techniques entails new security challenges in the detection of cyber attacks as it may hampers the detection of the attacks. Moreover, the costs associated with the deployment of 5G and 6G infrastructures for Radio Access Network (RAN) and transmission may increase from 60% to 300% [4]. Thus, solutions must be sought to reduce costs, both capital and operational, in the network infrastructure. This mission is hampered by the problems associated with the massive number of connected IoT devices, which expose significantly higher vulnerable perimeters for possible DDoS attacks. An example is the one that happened repeatedly in 2021 on the Voipfone Digital Service Provider (DSP) in United Kingdom, which suffered from this type of attacks between September and October, leaving no VoIP services to companies that used their products [5]. They indicated the impact caused by such attacks on companies: "if businesses are deprived of their services, they are deprived of business". This means not only capital loss but also loss of confidence in their services. One of the last Cloudflare's DDoS report [6] show that network-layer DDoS attacks increased by 109% in 2022 Q2. They also state that for those network-layer threats, attacks Telecommunication companies grew a 66%.

Fig. 1 presents an approximation of the architecture of a 5G/6G system (5GS/6GS) used to connect IoT networks, consisting of four layers. The IoT Device layer is composed by smartphones and other IoT devices, connected to the corresponding Radio Access Network (RAN) layer provided by the base stations (e.g., 5G gNBs). These gNBs are a set of physical and virtualised components, starting with the Distributed Units (DUs) [7], which support the lower layers of the protocol stack such as Radio Link Control (RLC), Medium Access Control (MAC) and physical layers, and ending in the Virtualised Central Units (vCUs) [7], which support different protocols such as Service Data Adaptation Protocol (SDAP) [8], Radio Resource Control (RRC) [9] and Packet Data Convergence Protocol (PDCP) [10]. The RAN layer then connects to the Edge layer through the CUs as they are virtualised and deployed in the Mobile/Multi-access Edge Computing (MEC) Network [11–13]. The aim of this layer is to bring the functionality and computing capacity geographically closer to the end user. Therefore, in addition to network functions, other applications that could be categorized as MEC can also be brought closer. The 5G/6G Core layer provides functions such as Session Management Function (SMF), User Plane Function (UPF) and Access Management Function (AMF). These functions have been virtualised. As in the Edge layer, cloudified functionalities in the Core layer can offer Cloud as a Service applications, running in the core network of the infrastructure with a greater computational capacity and a centralization of resources.

The 5G/6G System (5GS/6GS) architecture presents different stakeholders that are involved in the provisioning of network resources, as presented in the View on 5G Architecture by the 5G Public Private Partnership (5G PPP) [14]. A major role in the provision of 5G/6G services is the Digital Service Provider (DSP), which offers digital services such as enhanced mobile broadband and IoT to various vertical industries, and the role of (Virtual) Infrastructure Service Providers (ISPs) offering infrastructure as a service.

Either a DSP or an ISP can implement individual closed control-loops without any human intervention in order to fully automate the response against cyber-attacks in their respective administrative domains. If, in fact, both stakeholders simultaneously do so, each system will work completely independently focused on the protection of the infrastructure of its respective owner. However, it is important to emphasize that all the traffic that is being received by the DSP has necessarily trespassed the ISP domain as the DSP is embedded (virtualized) inside of the ISP domain. Thus, even if both systems work fully independent, the automated cyber-security loop of the DSP may perceive the actions done by the ISP loop and vice versa due to the side-effect that their respective actions cause in the system of the other stakeholder. Thus, if the ISP cyber-security system stops a malicious flow that was intended to transverse the DSP, then as a side-effect the DSP cyber-security system will perceive that such malicious flow

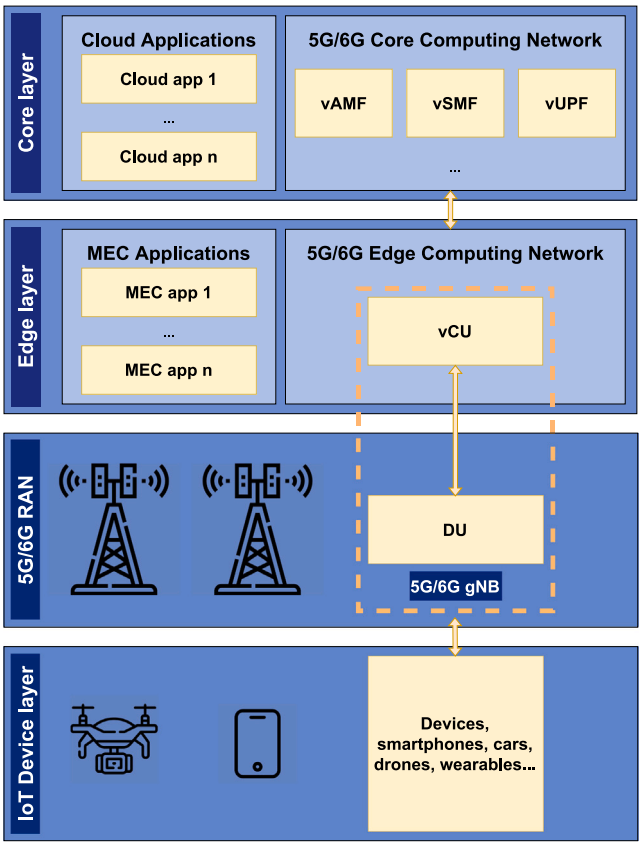

Fig. 1. IoT architectural layers coupled with the 5G architecture.

is not existing anymore (although it will never know the reason for that to happen). Notice how this side-effect causes that they end up collaborating to perform a more effective and efficient mitigation of the attacks even if the two loops operate independently on their respective administrative domains. This is the main hypothesis investigated in this manuscript. The benefits of the dual closed control loop proposed in this paper is of paramount importance for each of the stakeholders. If the mitigation system only operates in one of the stakeholders (ISP or DSP), it would be fully responsible for keeping the integrity and availability of the system in the event of an attack, and it could be saturated when the volume of the attack reaches exacerbated proportions. However, by placing such a system in both stakeholders, they will act independently for the mitigation of the same attack, mitigating parts of the ongoing attack and sharing the computational load without even the need to communicate between each other. This means that the DSP will mitigate part of the attack on its infrastructure while the rest of the attack that may go unnoticed will be mitigated by the ISP due to independent operations in their own administrative domains. The same happens in the implementation of a multi-tenant architecture, where each of the tenants, together with its corresponding host, will act cooperatively to mitigate the same attack.

The main contribution of this research is the design, prototyping and validation of a distributed dual-layer cognitive dual loop system to self-protect a multi-stakeholder multi-tenant 5G-IoT infrastructure from DDoS attacks without human intervention. The following contributions and innovations are achieved in this work.

- Autonomous DDoS mitigation system suitable for 5G/6G networks without collateral damage has been achieved through the development of a complex autonomous system, including detection of malicious flows and triggering fine-grained alerts, analysing the attack and determining a plan of intervention,

Table 1
Challenges to achieve distributed dual-layer self-protection closed cognitive loop for IoT networks against DDoS attacks.

Group	Code	Description
Stakeholders	I	Supported data and management plane for Infrastructure Service Provider (ISP) stakeholder
	II	Supported data and management plane for Digital Service Provider (DSP) stakeholder
	III	Supported collaboration between stakeholder Infrastructure Service Provider (ISP) and Digital Service Provider (DSP)
5G System involved	IV	Traditional IP traffic
	V	Multi-tenant Overlay Network (MON) with VxLAN/GRE (Cloud Infrastructures)
	VI	5G/6G IoT device traffic over a tenant
	VII	5G/6G IoT device Mobility across Gateways
	VIII	Geographical distribution of machines (Edge & Core)
Cognitive loop capabilities	IX	Accurate detection of the attack
	X	Analysis of the threat detected
	XI	Decision generation to be taken in order to mitigate the threat
	XII	Planning of the decision to be taken
	XIII	Orchestration of the plan to execute
	XIV	Actuation to mitigate the current attack
	XV	Enforcement of the actuation in the overlay networks
Others	XVI	Real environment (a), emulated (b), simulated (c)
	XVII	Type of attack: UDP (u), TCP (t), HTTP (h), Signalling (s), N/A (-)
	XVIII	Centralized mitigation (cm), Distributed mitigation (dm), Both (b), N/A (-)

orchestration that arranges the assembled plan and launching the plan at a predetermined time and location.

- Collaborative DDoS mitigation system between two of the stakeholders, achieving a dual concurrent closed control-loop. The first control loop runs in the ISP concerned with defending its own physical infrastructure and maintaining isolation and continuous operation of its services. The second loop runs in the DSP responsible for the management and safekeeping of its Virtualised Infrastructure Services and Digital Services.
- Distributed DDoS mitigation system involving all the locations that are exposed as part of the perimeter of the infrastructure instead of the traditional centralized approach where the enforcement point is carried out in only the central location.

The rest of the paper is organized as follows. Section 2 exposes the related work with different approaches to the detection, analysis, and mitigation of DDoS attacks in IoT and/or 5G IoT environments. Section 3 describes the approach of this contribution by showing the design and architecture towards a self-managed protection for IoT networks. Section 4 describes in detail the flow of the contribution in this work, detailing how the collaboration between each of the stakeholders is done. Section 5 presents the evaluation and performance of the proposed architecture and system. Section 6 concludes the paper together with future work.

2. Related work

Table 2 shows a summary of the contributions that have been analysed to represent the state of the art of this work. Key relevant characteristics have been established and compared to give an overview of the main contribution of our contribution. In order to have a readable table, the column names have been explained in Table 1 to allow the reader to better understand the analysis carried out.

In [15] a framework is presented that allows the detection of malicious network traffic at the IoT-Edge layer and thus identify possible infected IoT devices in a Botnet network. The analysis is carried out using Sparsity Representation and Reconstruction Error Threshold techniques, which allows it to recalibrate the decision point on which the traffic is being classified. The dataset used to train the ML models is the NB-IoT and only benign traffic data is used to calculate the threshold error. This framework however is not presented to act in across all layers of the 5G/6G architecture and its effectiveness for multi-stakeholder environments has not been studied either. In addition, the authors have not consider the classification of 5G/6G traffic, then resting credibility on the validation of the approach.

[16,17] present similar work in terms of the characteristics represented in this publication. Both perform attack detection and subsequent analysis of the attack, in order to take action and mitigate the attack. [16] specifically presents IoT Botnet Detection and Analysis (IoT-BDA), a framework for detecting, analysing, identifying, and reporting botnets circulating on the Internet. The framework consists of two main blocks: Botnet Capturing Block (BCB), which is composed of a set of pre-prepared honeypots with vulnerabilities that can be exploited by botnets. The BCB will continuously report the activity that is occurring on its system to the next action block of the framework. Botnet Analysis Block (BAB), which consists of an API, parsers, sandbox, analysers and reporters. It executes or replicates the actions of the honeypots on the sandbox so that the analysers, using anti-malware programs and techniques, can detect whether the traffic being generated comes from a botnet or not, as well as applying analyses to discover whether these botnets use anti-analysis techniques, anti-forensics, etc. Once a potential botnet has been identified, it is reported. On the other hand, [17] presents a framework which acts on the firmware of IoT devices. It tries to use Deep Learning (DL) techniques by using the Long Short-Term Memory (LSTM) algorithm in order to detect the attack that is happening through its network. To mitigate the attack, it analyses the infected device and disables it. The framework continues to listen and re-enables the devices to confirm whether or not they are still infected and, if so, disables them again. Although the above contributions are of great interest, they do not present a solution that can perform the mitigation of attacks in the overlay networks available in the 5G system.

[18] presents a two-level DDoS attack detection method based on information entropy and deep learning for Software Defined Networking (SDN) environments. The first level is based on the entropy detection mechanism detects suspicious components and ports in coarse granularity. The second one executes a fine grain packet-based detection by a Convolutional Neural Network (CNN) model to distinguish normal traffic from suspicious traffic. In the end, the controller is in charge of the mitigation of the attack by the interception of it. They use the open dataset CICIDS2017 to train their model and gather deep insights of the complete analysis they do for their work. The experimental results show that the two-level detection mechanism has high detection accuracy and efficiency. The above contributions are limited to traditional IP networks, without getting involved in the complexity of the overlay networks presents in 5G multi-tenant systems.

[19,21–23] present works that, in addition to obtaining an accurate detection of the attack for subsequent analysis, they perform their tests on simulated environments for 5G systems, allowing the execution of their work on the Network Operators' infrastructure. [19] presents a framework capable of detecting Silent Call Attack, Signalling Attack

Table 2
Analysis of the state of the art with respect to our contribution against the challenges required to achieve distributed dual-layer self-protection closed cognitive loop for IoT networks against DDoS attacks.

	Stakeholders			5G/6G System involved					Cognitive loop capabilities							Others		
Reference	I	II	III	IV	V	VI	VII	VIII	IX	X	XI	XII	XIII	XIV	XV	XVI	XVII	XVII
Tzagkarakis et al [15]	-	-	-	-	✓	-	-	-	✓	✓	-	-	-	-	-	c	-	-
Trajanovski and Zhang [16]	-	-	-	✓	-	-	-	-	✓	✓	-	-	-	✓	-	c	h	c
Salim et al [17]	-	-	-	✓	-	-	-	-	✓	✓	-	-	-	✓	-	c	-	c
Liu et al [18]	-	-	-	✓	-	-	-	-	✓	✓	-	-	-	✓	-	c	u	c
Hussain et al [19]	✓	-	-	✓	-	-	-	-	✓	✓	-	-	-	-	-	c	s	-
Baig et al [20]	✓	-	-	✓	-	-	-	-	✓	✓	-	-	-	-	-	a	t	-
Silva et al [21]	✓	-	-	✓	-	-	-	-	✓	✓	-	-	-	✓	-	c	s	c
Liu et al [22]	✓	-	-	✓	-	-	-	-	✓	✓	-	-	-	✓	-	c	h	c
Li et al [23]	✓	-	-	✓	-	-	-	-	✓	✓	-	-	-	-	-	c	-	-
Candal Ventureira et al [24]	✓	-	-	✓	-	-	-	✓	✓	✓	-	-	-	✓	✓	c	-	c
Paloalto [25]	✓	✓	-	-	✓	-	✓	✓	✓	✓	-	-	-	✓	✓	a	s	c
Serrano et al [26]	✓	-	-	✓	✓	✓	✓	✓	✓	-	✓	✓	-	✓	✓	b	u	b
Our work	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	b	u	b

and SMS Flooding Attack DDoS attacks in the communications infrastructure for 4G LTE-A architecture. This detection is carried out by a DL Convolutional Neural Network, which analyses in a preprocessing of the traffic in the CORE of the network using ResNet-50 model to finally classify the type of traffic and discern from one attack or another. To train the DL model they use an open dataset from Telecom India. [21] presents REPEL, a new framework capable of detecting signalling DDoS attacks for 5G environments but focusing on prevention by scaling virtualised network services. They do not explaining the architecture of their Intrusion Detection System (IDS) and instead focus their arguments on the load balancing associated with prevention and action strategies in the 5G control plane, as well as proposing an attacker obfuscation system. [22] presents Umbrella, a defence mechanism against DDoS attacks which is deployed at the ISP, thus acting at the network level. They claim that, thanks to their multiple layers of work, they can stop different types of DDoS attacks at the network level, however, they depend directly on the continuous response of the victim, which must provide a list of trusted IPs that it has in its history. Finally, [23] propose an IDS based on Machine Learning (ML) techniques in 5G systems for Software Defined Networks. The architecture of this framework consists of three different abstraction layers, which are: forwarding, where network traffic is generated by OpenFlow-controlled entities (OFs); Management and Control layer, in charge of flow and packet collection as well as anomaly detection; data and intelligence layer, where ML algorithms are executed for the analysis and classification of traffic that has been detected as anomalous. On the other hand, [20] presents an interesting approach to a framework capable of the detection of DDoS threats in IoT networks. They analyse the performance of six different classifiers, A1DE, A2DE, Naïve Bayes, Bayesian Network, C4.5, and MLP. Finally they conclude that their ADE-based Denial-of-Service (DoS) attack detection scheme demonstrate a good performance for the different experiments that it has been tested with. The main difference with our contribution is observed in the scenario where the tests were carried out, in addition to the complexity of the task once the attack was detected. They have not used a test environment consistent with reality but have used simulations to be able to approximate a possible solution to a simulated problem.

In the article [24] a framework capable of detecting IoT devices susceptible of being infected for 4G and 5G networks is presented. The attack mitigation is performed by placing the traffic of these devices in quarantine on a Network Slice (NS) dedicated to this task, where it will then be analysed in depth to classify it as a malicious attack. This detection is carried out by an application running on top of the SDN controller, which will generate a distrust threshold as the number of flows in the quarantine NS varies. However, this contribution lacks of mitigation capabilities in the DSP, being limited only to ISP, as opposed to ours contribution that focuses on the collaboration of all stakeholders involved in the infrastructure for the protection of its physical and virtual resources.

Paloalto Networks [25] offers Strata, a next-generation physical firewall that enables a security layer to be applied to 5G Network Operators to act against signalling attacks. It enables this network security to be applied across the 5G infrastructure, providing user mobility through the GPRS Tunnelling Protocol (GTP) encapsulation. It allows the network engineer to standardize different NS to be able to classify the traffic that is occurring along the infrastructure, in addition to being able to apply different control rules on the flows.Although it is one of the closest solutions to ours contribution, [25] does not present a clear solution for the entire 5G infrastructure, with only the CORE infrastructure being protected and not EDGE.

Serrano et al. [26] have demonstrated successfully a close control-loop suitable for performing the self-protection of 5G networks. Our approach follows the same approach where we have significantly extend that work with new capabilities to perform parallel cognitive control loops involving different stakeholders.

It has not been possible to find in the literate any example of concurrent close control-loop for self-protection of infrastructure where both ISP and DSP loops are executed as a way to establish a collaboration between them. This lack of results in this critical area of security has motivated the writing up of this manuscript.

3. Towards a self-managed protection architecture for IoT networks

The system proposed in this work is explained in detail in the following sections. Firstly, an introduction to the 5G multi-tenant networks, which are an essential part of the proposed system to achieve the objectives mentioned in Section 1, is provided. Secondly, a more detailed explanation of each of the software components of the self-protection control loop proposed as a contribution to this work and the innovation they entail is explained, and these are described in different subsections: Security Monitoring Agent (SMA), Analyser, Decision Maker, Planner, Orchestrator and Flow Control Agent (FCA).

3.1. 5G multi-tenant network traffic

As explained in Section 1, there are several stakeholders involved in a 5G/6G system, all of them with different business models to satisfy their customers. This is why each of these stakeholders will have a different purpose in terms of acting on the data flow that is happening in their infrastructure (either physical or virtualised). This means that, along the entire route that a data flow must take, from its creation in the IoT device, to its destination, passing through the entire 5G/6G communications infrastructure, it will have variations in its morphology, despite its invariability in content. To achieve this, various encapsulation mechanisms are used to isolate traffic from each of the tenants that may be installed on the same infrastructure. Virtual Extensive LAN (VxLAN) [27] is one of the examples, which is used to create the so-called multi-tenancy, isolating the traffic for each of the tenants which are occupying the infrastructure. This technology allows to create a first encapsulation of the data flow to achieve such isolation which will be used to identify the tenant. GTP is used as a second encapsulation, according to the standards of the 5G and expected in

the forthcoming 6G mobile networks, allowing end-user mobility. This is described in detail in [28] in case the reader is interested.

Our proposed architecture is validated against 5G/6G network traffic in order to meet the associated requirements and challenges, such as the feasibility of multi-tenancy, self-adaptation to topology variations and mobility. To deploy this architecture, it has been followed an approach as described in this paper [29] where a Hybrid and Extensible architecture is run to achieve ISP and DSP deployment.

Fig. 2 shows the architecture of the 5G-IoT network used in our research work. It shows the different domains where the network intervenes, starting with the IoT devices in the Device layer (see 1 in Fig. 2). These IoT devices send the messages, in our case a raw DoS attacks, to the 5G RAN layer, where the antennas, together with the User Equipment devices (UEs), route the traffic (see 2 in Fig. 2) to their respective Edge Computing infrastructures. The DUs have been tasked with using the Common Public Radio Interface (CPRI) protocol to encapsulate the raw data from the devices so that it can be correctly routed by the ISP's physical devices. Once traffic arrives at the Edge Computing layer (see 3 in Fig. 2), it is redirected to the DSP in concerned by the first virtual switch it encounters (represented by a purple rectangle). This DSP, boasting the functionality of a vCU plus MEC capabilities, is responsible for non-real-time, higher L2 and L3 (network layers) and stack functions (see 4 in Fig. 2). When the packet leaves the DSP (represented by a yellow rectangle), it is already encapsulated in GTP and passes through a final virtual switch at the ISP to be encapsulated by the VxLAN protocol on its exit (see 5 in Fig. 2). On leaving the ISP in the 5G Edge Network, the packet is routed directly to the central physical switch/router in the Transport Network (see 6 in Fig. 2), where it will be routed to the appropriate ISP in the 5G Core Network. When the packet arrives at the ISP in the 5G Core Network, it comes with all the above encapsulations (see 7 in Fig. 2), so the first step that the very first virtual switch it encounters is to undo the VxLAN encapsulation that the ISP in the 5G Edge Network has performed. Again, the packet is sent directly to the corresponding DSPs (see 8 in Fig. 2), where thanks to the GTP encapsulation it is known which one it belongs to. The relevant tasks are performed in the DSP, the GTP encapsulation is undone and the packet is sent back to its destination, passing through the last virtual switch of the ISP where, if the packet is inspected, it can be seen that only the raw (IP) information sent by the UE in the first instance remains, with the information referring to its destination (see 9 in Fig. 2). Both in the infrastructure provided by the ISP and the DSPs, all traffic is being replicated by a switch that acts as a mirror to the service layer (see 10 and 11 in Fig. 2) where the first component of the protection loop discussed in this paper, the Security Monitoring Agent (SMA), is installed so that, in case an alert is registered, it will communicate with the Management layer (see 12 and 13 in Fig. 2) where the alert will be analysed and the appropriate steps described in the following Section 3.2 will be executed. Finally, the packet is sent to its destination through a switch that routes its traffic to the internet (see 14 in Fig. 2) in order to reach its destination server. It is important to mention that VxLAN traffic simply does not exist from the point of view of the DSP as all its services will never see such protocol and thus the mitigation of traffic done by the DSP will be completely different in terms of the packet structure from the one carried put by the ISP due to the different encapsulations available in each point of the infrastructure.

3.2. Self-protection close control-loop components

The self-protection close control-loop presented in this paper is the composition of different software components that are architecturally chained to conform a close loop. Each of these components has a specific task and the combination of all of them results in the accurate detection of an attack, the analysis of its cause and the subsequent action against the threat. The communication between each of the components is done by a Message Bus software. This capability grants communication for the components through a publishing/subscription architecture. Here is where the concept of "Exchange" or "Topic" is used to describe where the information is published and where the components must subscribe to receive such information. The current implementation supports two different Message Bus technologies: Apache Kafka [30] and RabbitMQ [31]. For this work, it is been considered to use only RabbitMQ as communication technology between the self-protection loop components.

As an introduction, the tasks and functionalities of each of the loop components are explained in the following subsections, giving an overview of the role of each of these components in the system, their responsibilities and their impact on the system in order to justify their participation in the loop.

3.2.1. Resource Inventory Agent (RIA)

The Resource Inventory Agent (RIA) is a component responsible for providing information about the topology of all network devices, ports and connections between ports and devices available on each machine. The information that RIA discovers in order to supply it to the rest of the components is: (i) Physical machine and its logical and virtual network interfaces, (ii) VMs and their virtual network interfaces, (iii) Containers and their network interfaces, (iv) Software switches, (v) Specialized physical devices such as Software Defined Radio (SDR) and their network interfaces, (vi) Interconnection between network interfaces, and (vii) Multi-tenant information of VMs. This component is also instantiated in all the machines of the infrastructure: Service and Compute layers of each Edge and Core segments and for each stakeholder. The capabilities and performance of this component are detailed in [32], and a specific use case for the topology information that the component discovers is specified in [33].

3.2.2. Security Monitoring Agent (SMA)

The SMA component has been developed with the main objective of enhancing and extending the capabilities provided by a traditional IDS. This is due to the fact that the capabilities of the traditional NIDS lack of 5G infrastructure and network information. In that case, this SMA has the responsibility of the fluent communication with the NIDS and the addition of relevant metrics and information of the 5G infrastructure regarding to the malicious flow that the NIDS has alerted. The traditional IDS that is being used for the purposes of this work is Snort [34]. The SMA is the combination of this Snort IDS together with a 5G multi-tenant traffic classifier created by us that allows the collection of relevant information about the 5G network tenants and allows the generation of concrete, granular and effective alerts. This fine-grain alerts allow to minimize the collateral damage of the attack as, they allow to stop only the malicious flows without affecting to the legitimate ones. The information provided by this component is separated in three categories differentiated by their purpose: first, metric information of the NIDS, such as the number of total packets of a specific technology that have been filtered or the percentage of the dropped packets; second, information about the malicious flow relative to the network information; and third, metadata information of the specific topology discovered by the RIA component that matches with the malicious flow structure. The capabilities and performance of this sensor are detailed in [35].

3.2.3. Analyser

The Analyser is the first component deployed in the management layer. Its purpose is to analyse the metrics that the previous component (SMA) has reported to the Metrics Exchange. This analysis is performed using the spatial and temporal information provided as metadata together with the metrics reported by the SMA. To have a better understanding of the purpose of this component, it is necessary to state that the NIDS with the SMA are reporting tons of flows produced by the DDoS attack. In that case, the system would only need to stop the specific flow that is causing the attack and not the repeated ones

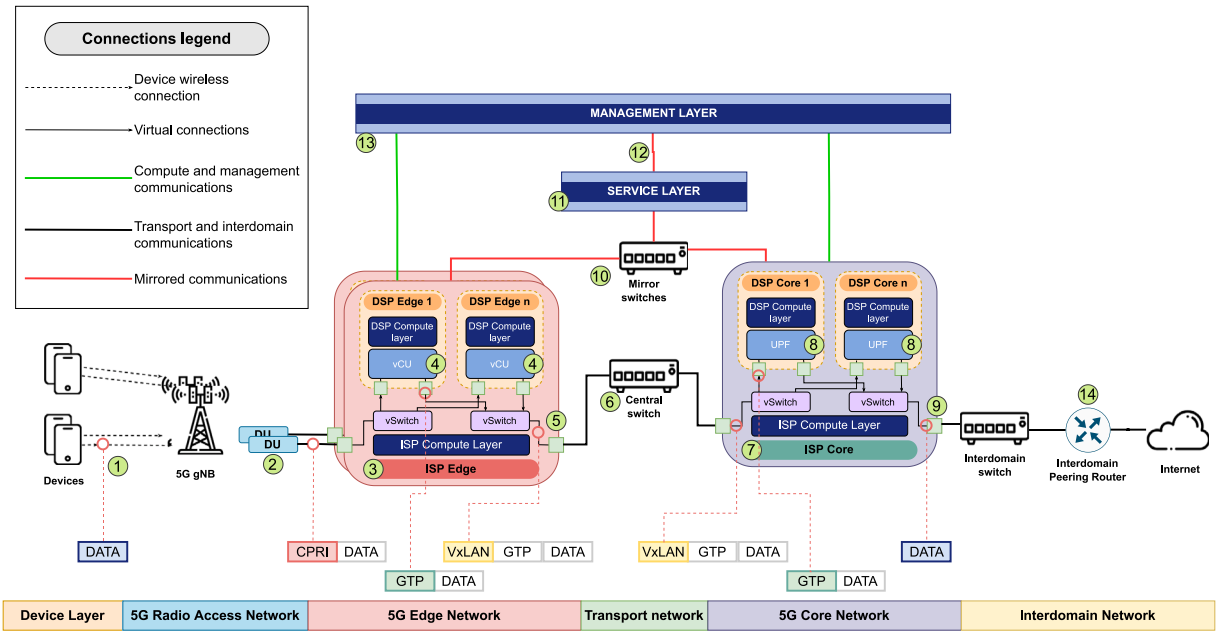

Fig. 2. Flow example of the traffic occurring in the 5G System infrastructure.

that might be being reported to the Analyser. Therefore, the Analyser executes its logic to retrieve the specific information of the malicious flows gathered by the SMA and specifies additional information such as: alert type, alert impact, causes of the alert and if the flow is already stopped or not. This analysis is intended to be reported in the form of an Alert to the Alert Exchange when there is something detected from the reported metrics and metadata. Such Alert will be consumed by the next component of loop, the Decision Maker.

3.2.4. Decision Maker (DM)

The Decision Maker is the component responsible for generating a decision. This decision is asserting what action should be taken and where. The actions can be diverse, such as performing a drop, redirecting traffic or mirroring the flow. On the other hand, the location on where should be enforced is determined following logic predefined by the programmer, such as "near the source", "near destination", "n hops from the source", etc. These actions can be automated by the administrator through a policy engine in order to define different strategies for each type of attack. The decision is taken by the analysis of the alert triggered by the previous component in the loop, the Analyser. In the case of this work, the decision that is executed has as a DROP action in the closest place to the source of the attack that is happening through the network.

3.2.5. Planner

The Planner component is responsible for generating a plan. This plan is the extension of the information provided by the Decision in order to achieve a set of implementable actions into the real system with the intention to complement any missing information not indicated in the decision with aspect such as: "default duration", "default way to perform the action", etc. These set of steps define precisely what action to take, where to take it, how to take it, how long to keep it active and when to take it, among others. Furthermore, this component is in charge of the computing of this "close to source" location where the attack must be stopped. This is made possible by the information that the RIA component is periodically reporting about the 5G network infrastructure and topological information. The Plan is published to the Plan Exchange.

3.2.6. Orchestrator

The Orchestrator component is in charge of executing the plan previously organized by the Planner. This is because the plan may contain a set of actions to be performed at different times and locations, as well as different points of action for the same attack that is happening throughout the network. The Orchestrator is the last component deployed in the management layer. Thus, additionally, the Orchestrator has the responsibility, which is to route the messages so that they can reach their previously indicated destination, i.e., each of the actuators deployed in the infrastructure are to receive the step of the plan that is being executed at that moment. Each step is published to the Intent Exchange and will be consumed by the FCAs that have been installed along the network infrastructure as explained in the next sub-section. Thus, different FCAs across the network will be responsible of enforcing different actions depending on the location they are instantiated in.

3.2.7. Flow Control Agent (FCA)

The FCA component is an agent whose main function is to expose network traffic control. Each of the computers that are present in the infrastructure have an FCA agent associated and installed which enables the control of the network traffic that exists between its physical and virtual machines. It differs from other agents such as SNMP and OpenFlow as it is an abstraction layer on top of different control technologies such as iptables, OpenFlow, SNMP, Traffic control (tc), etc., capable of exposing functionality to the management plane. The FCA component is able to provide distributed mitigation as it is multi-instantiated across the whole infrastructure and it will serve as an API with the protocol compatible with the different implementations in the data path. The capabilities and performance of this component are detailed in different manners in the following citations [36–38]. As described in the previous component (Orchestrator), the FCA will be instantiated in the different Compute layers across the 5G infrastructure. It will receive the organized information provided by the Orchestrator in order to enforce the actions in the dataplane. For the purpose of this work, the control technology that is being used in the experiments is tc.

4. Collaboration between software components

To clarify the relevance of the intervention of each of the components in the self-protection system, it is necessary to study Fig. 3, which

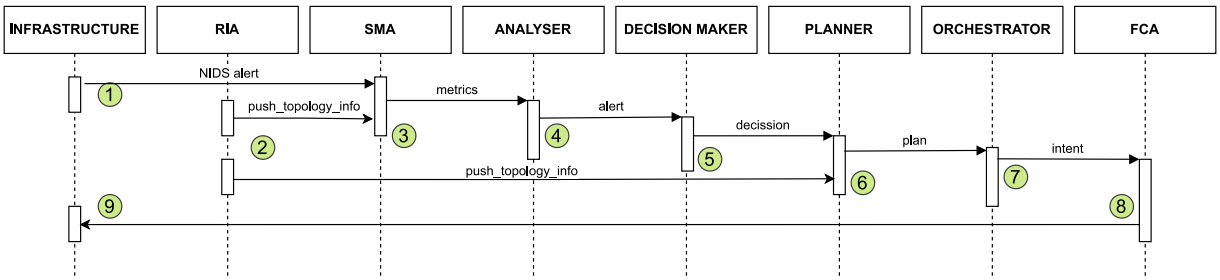

Fig. 3. Sequence Diagram for the self-managed protection loop.

represents the sequence diagram that the system follows to achieve the goal: to protect the infrastructure (physical and virtual) in an autonomous and collaborative way. As can be seen in the diagram, the beginning of the loop takes place in the infrastructure of the scenario to be monitored (see 1 in Fig. 3), where the IDS has detected the threat and reports it. At the same time, the RIA component has also reported the topology information so that the rest of the associated components can perform their task (see 2 in Fig. 3). Next, the SMA component, which has collected the Network Intrusion Detection System (NIDS) alert and the topology information reported by RIA, will do its analysis and collection of metrics that it will end up reporting (see 3 in Fig. 3). These metrics are collected by the Analyser component, which will create the appropriate alert for reporting (see 4 in Fig. 3). The alert is picked up by the Decision Maker component and formulates the decision to follow to mitigate the threat that is happening in the system (see 5 in Fig. 3). The Planner component comes into action when it collects the decision made by the previous component and unifies this information with the information also collected by the RIA component. It creates a plan based on the topology information and the decision against the existing threat and finally reports the plan (see 6 in Fig. 3). This plan is picked up by the Orchestrator component, which will create the list of plans to follow and report them as an Intent (see 7 in Fig. 3). Finally, it is the active FCA component that applies the intent in the form of a rule at the place in the infrastructure that has been chosen by the Planner component (see 8 and 9 in Fig. 3).

Fig. 4 depicts the different planes on which the protection loop operates, as well as a clear distinction between the stakeholders involved in the infrastructure. Notice how the scenario has multiple separated management planes, each per each of the DSP present in the architecture plus an additional one for the ISP. This allows to run parallel control loops where each of the loop is in charge of providing self-protecting capabilities to the portion of the infrastructure they are in charge of. The figure allows the reader to understand that the reaction of one of the management planes will have side effects in the behaviour of the other management planes due to the fact that the traffic transverses across all of them. Let us take this Fig. 4 as an example to explain the flow that the self-protection loop is using in case of an attack. Assuming a potential attack that is passing through the network to the internet, it will be redirected (as well as the legitimate network traffic) to the mirror switches that are connected to the specific stakeholders' Service layers (see 1 in Fig. 4. Here, the dataplane is being red by the first component, the SMA that is capable of the detection of this attack (see 2 in Fig. 4. It is important to note that the encapsulations of each flow affect to the performance of the SMA; if there are more nested encapsulations (i.e. VxLAN over GTP over CPRI) the analysis of the information will be slower. Also, as the traffic is being mirrored to each of the stakeholder's Service layer with the only information that they can only gather (ISP will be able to get VxLAN encapsulations but not DSP). Here starts the collaboration without exchange of any information between stakeholder, because each of them are only working with information they can gather on their own. The information provided by the SMA will be published to its messagebus exchange and the following component (Analyser) will make use of it in the Management layer, where 4 of the rest of the components take place (see 3 in Fig. 4). Once the Orchestrator finishes its task, the FCA in the specific Compute layer for the specific stakeholder will be in charge of enforcing the action to be taken in order to protect the infrastructure (see 4 in Fig. 4. In the case of a DSP, it will not be able to enforce a protection rule in the ISP infrastructure, nor the ISP in the DSP infrastructure. However, whether the attack is mitigated by any of the stakeholders, the goal of this contribution will be accomplished and the resources will be healed and protected, without interchange of any kind of acknowledge or information between the stakeholders and leaving the tightness of each network operator unspoiled.

5. Validation

To demonstrate the functionality of the system described in previous chapters of this work as a novel innovation, a testbed has been developed in which a 5G IoT network topology has been emulated and exposed to a series of DDoS attacks varying its volumetry. Two substantially opposite scenarios have been compared to test the effectiveness and benefits of the proposed system: the first scenario is based on the installation of the self-protection loop in a single stakeholder of the 5G IoT network, i.e. in the ISP; the second scenario is the contribution of this work itself, where the loop is installed in each of the management layers of the stakeholders involved in the network, creating a distributed dual-layer self-protection loop of the network.

5.1. Testbed and experiments design

All the software components described in Section 3 have been designed, prototyped, deployed and validated in real 5G mobile edge computing infrastructures. They are all implemented in Java 17 with the only exception of FCA, which is implemented in Python 3. SMA uses Snort 3.0 underneath to perform the detection of attacks. RIA makes use of a collection of tools and mechanisms including OpenStack (Wallaby or higher), OpenAirInterface 5G (W44 2022 or higher), LLDP, CDP and iproute2 (v1.9 or higher) Linux stack to detect the topologies. We employ RabbitMQ 3.6 as Message Broker and MySQL 8.15 as DB. Analyser, Decision Maker and Planner are based on a MySQL 8 engine as a way to allow the usage of SQL to express Analytical policies, Decision Making policies and Planning policies. The orchestrator is a Java implementation and FCA relies on OpenVSwitch 2.17.3 and iproute2 (v1.9 or higher) to enforce the mitigation actions.

To validate the effectiveness of the contribution of this work, Common Open Research Emulator (CORE) [39] has been employed as the emulation tool of network topologies, which makes use of Linux Network Namespaces (netns) to emulate each of the different devices and networks on the infrastructure, being able to share the same file system and kernel, but generating their own private network and process environments. This, combined with the Linux Ethernet bridging tools provided by the Linux environment itself, allows any type of network to be emulated, including the wireless ones needed to faithfully represent the infrastructure detailed in this publication. The physical machine used in this work generated a virtual machine, on which the

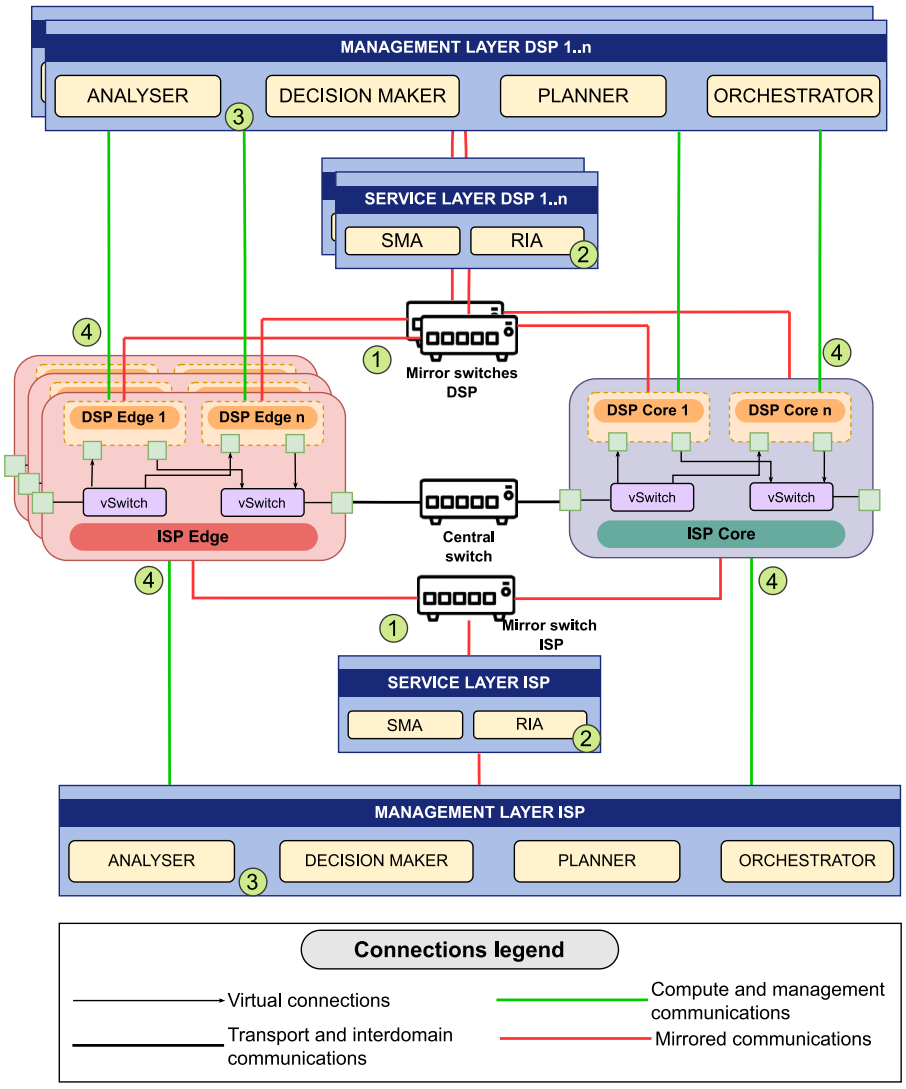

Fig. 4. Cognitive Self Protection Loop components relationships.

experiments were run, with a Linux operating system, in its Ubuntu 16.04 LTS distribution and kernel 4.4.0. The 10 cores of the Intel Xeon(R) CPU E5-2630 v4 and 16 GB of RAM have been added to this virtual machine. The CORE Emulator is a proven tool remarkably like Kubernetes, Docker Compose or OpenStack but it has the additional capability of allowing describing any topology in a graphical interface in a drag-and-drop fashion. Therefore, we have decided its usage in this experimentation. Notice that under any of such icons presented in Fig. 5, there is a completely functional Ubuntu 16.04 Linux distribution.

CORE is one of the projects with the greatest vision in the field of academic research in networks and communications, and that is why it offers a wide range of tools and possibilities to develop each of the projects that are to be carried out with it. This facility has allowed the development of a system for the creation, configuration, provisioning, emulation and execution of different experimental scenarios. Fig. 5 shows the CORE Graphical User Interface (GUI), where a loaded scenario of one of the experiments launched in this work is shown. This experiment consists of 32 infected IoT devices, which are represented under the orange discontinuous box, the User Equipment (UE) label, which in turn, are connected to their respective antennas (5G gNB) and these leave their Devices and Radio Access Network domain (see 1 in Fig. 5) and route their traffic to the corresponding DSP in their EDGE Network (see 2 in Fig. 5). As can be seen, there are several antennas connected to each EDGE Networks according to the spatial topology of the network, being represented within the green squares and indicated by an EDGE-1 and EDGE-2 label. In turn, traffic is routed to the Core network through the transport switch (see 3 in Fig. 5), represented as a purple square containing the machines acting as ISPs and DSPs (see 4 in Fig. 5). All traffic is mirrored to the service layer (see 5 in Fig. 5), represented as an orange square, where the first reactive component of the loop presented in this work (SMA) is located and that, once detected the flow that is considered potentially dangerous, starts to execute the self-protection sequence that is performed in the Management layer in each of the stakeholders, represented as the blue squares and annotated with MGMT-ISP, MGMT-DSP1 and MGMT-DSP2 labels (see 6 in Fig. 5). As the last part of the created infrastructure, there are the servers outside the network domain that act as victims and where the attacks are targeted (see 7 in Fig. 5). It is important to allow the reader to see how in both EDGE and CORE network segments the computers of the ISP are hosting different DSPs inside of virtual machines, creating a multi-tenant infrastructure.

The building of the scenario mentioned in the previous paragraph, consists of a sequence of steps in the creation of the environments executed by the CORE emulator so that the architectures and surroundings desired by the researcher can be emulated efficiently and the series of results can be collected. These steps are:

1. The CORE emulator loads the specified topology *IMUNES imn* file with information about name of nodes, links and connections

between nodes, interfaces and network information such as IP and MAC addresses. These nodes represent all computers, IoT devices, virtual machines and physical infrastructure that take place in the scenario.

2. Each node in the scenario has a role, and each role needs specific configuration to be run. The IoT devices will have the role of attackers in the scenario, so at this step they will be provisioned with their specific information to launch the DDoS attack. This information is about the target ip they are focusing, type of attack, bandwidth consumed, packets per second, etc. For the rest of roles: ISPs and DPSs will have information to create mirroring and tunnelling protocols, Compute layers and Service layers will get information regarding to the software components that need to be installed, and so on with the remaining roles.
3. For each experiment, the scenario will have different number of IoT devices, Compute and Service layers to instantiate, even different Edges per Core segments. This is why some configurations must be dynamically done in the automation of these steps. All nodes are provided by a global configuration at the beginning of the experiment, however, this configuration is re-written for each node (i.e. the network IPs allocation for each attacker or the mirrored interfaces to the Service layers when the number of Edge segments vary).
4. Each role has specific goal to accomplish and different software to run, so another step is necessary to install all software dependencies for each node depending on the software they are running. The Service layers will be provided by the dependencies necessary to run the SMA and RIA, as well as the Management and Compute layers with the specific dependencies to run the rest of software components that were described in Section 3.2.
5. After all nodes have their network and software configurations, it is time to deploy the desired encapsulations in order to achieve tenant isolation and user mobility. First, the DSP isolation is granted applying VxLAN tunnelling at the ISP interfaces as shown in 5 Fig. 2, similar to the OpenStack dataplane. The VxLAN encapsulation allows the ISP to redirect effectively the traffic that belongs whether a tenant or to another. The tool that is being used to create the VxLAN tunnelling in this work is Open vSwitch [40]. Second, the GTP encapsulation permits the user mobility and connects the IoT device to the destination server via TUN/TAP devices [41]. In this particular, the communication created is end-to-end and is the GTP server created with osmo-ggsn [42] software is responsible of the allocation of dynamic and ad-hoc IP for each device connected to it. This osmo-ggsn is compabilty with the dataplane of any 5G vendor such as Nokia, Ericsson and OpenAirInterface. Finally, it is selected which interfaces in the infrastructure are going to be mirrored to their respective Service layers. Whether the stakeholder is ISP or DSP, the mirrored interfaces will be done in ingress and egress mode to allow the detection of the direction of the malicious flow. With these 2 nested encapsulations and port mirroring the configuration step of the experiment finishes.
6. Once the configuration of the scenario and the experiment is done, the software components are started in order per each node. The Service, Compute and Management layers will run their software components and they will start listening to the traffic that is being mirrored. The scenario is ready to be tested and validated at this step.
7. Bonesi [43] is the tool installed in the IoT devices that creates botnet traffic with a chosen target IP. It can be modified to focus on different targets at the same time or to implement different types of DDoS attacks. For the purpose of this work, it will be launched a large UDP flooding attack from the IoT devices to the Servers. This UDP flooding attack works primarily by exploiting the steps that the server takes in order to answer UDP packets sent to one of its ports. As the server is receiving a huge amount of packets (deeper explanation is shown in Section 5.2) it will be overwhelmed and not able to respond to legitimate packets.
8. Finally, the experiment is completed and the results are gathered by the collector nodes installed in the Management layers of each DSP and ISP. The state of the machine is restored: cleans all temporary files of the experiment, deletes bridges that were communicating the host machine to the emulation environment, all created sub-processes are stopped, etc.

When all the infrastructure to be emulated has been created, configured and provisioned with the necessary software, the emulation of the DDoS attack by the IoT devices is carried out. The number of devices in the network has been varied in powers of 2, with a minimum of 4 attackers and a maximum of 256, which perform a UDP port flooding DDoS attack whose victim is a server that has been installed on a machine outside the network domain to be emulated. This is why all the attackers have each of the points of the network to be studied as a route, and it will be the network itself that effectively detects the attack, analyses it and determines a decision to act, orchestrates a plan and executes the appropriate actions to mitigate the attack that is happening as soon as possible in order to protect both its physical and virtual infrastructures. At all times, a series of software collectors that have been installed in the administration layer. They are collecting everything that happens in each of the components of the loop. This is why, at the end of the DDoS attack and once it has been correctly mitigated, these collectors finish gathering the necessary information to report it in this manuscript.

Since the main contribution of this work is to demonstrate the viability and benefits obtained by installing a self-protection loop (such as the one developed for this work) in each of the stakeholders involved in the network, it has been considered convenient that its functionality be directly compared with an opposite system, in which the same self-protection loop is only installed in the stakeholder that owns the physical infrastructures of the network. This system will henceforth be referred as a standalone system, comparing it qualitatively with the collaborative system offered as a innovation in this publication. The collaborative system is composed by 3 different cognitive control loops running simultaneously. One is running for the ISP and one is running for each of the two different DSPs deployed in the collaborative system. It has been designed to deploy two DSP in order to really show the multi-tenancy aspect where multiple tenants (DSPs) are sharing the same physical (ISP). This qualitative comparison of the performance of each system has focused on three key functional aspects: how effective the loop is in mitigating an attack, i.e., how many attackers the loop can stop out of all those that have been launched and whether it has managed to completely mitigate the attack; how long it takes to execute the self-protection loop, in order to know empirically how much overhead the self-protection system can withstand in the face of a massive DDoS attack; and as a third key point, studied along side the previous two, the responsiveness of the system when the number of attackers are being increased — with a minimum of 4 to a maximum of 256 in steps of powers of 2.

5.2. Functional results

The results shown here have been obtained by using the arithmetic mean on the indicator in question to be analysed. This is due to the fact that each experiment has been run 15 times and, while maintaining the same performance trend and times in each of the runs, there are slight changes in the results.

The first indicator to be studied is the average execution time of the self-protection loop. This is presented in Fig. 6, where a two-bar graph is shown. The green-coloured bar represents the average loop execution time for the standalone system, together with its trend curve of the same colour. On the other hand, the blue bars represent the result for the collaborative system, together with its trend curve of growth as the number of attackers increases. It can be seen how, for a small number of attackers, the standalone system seems to behave more efficiently

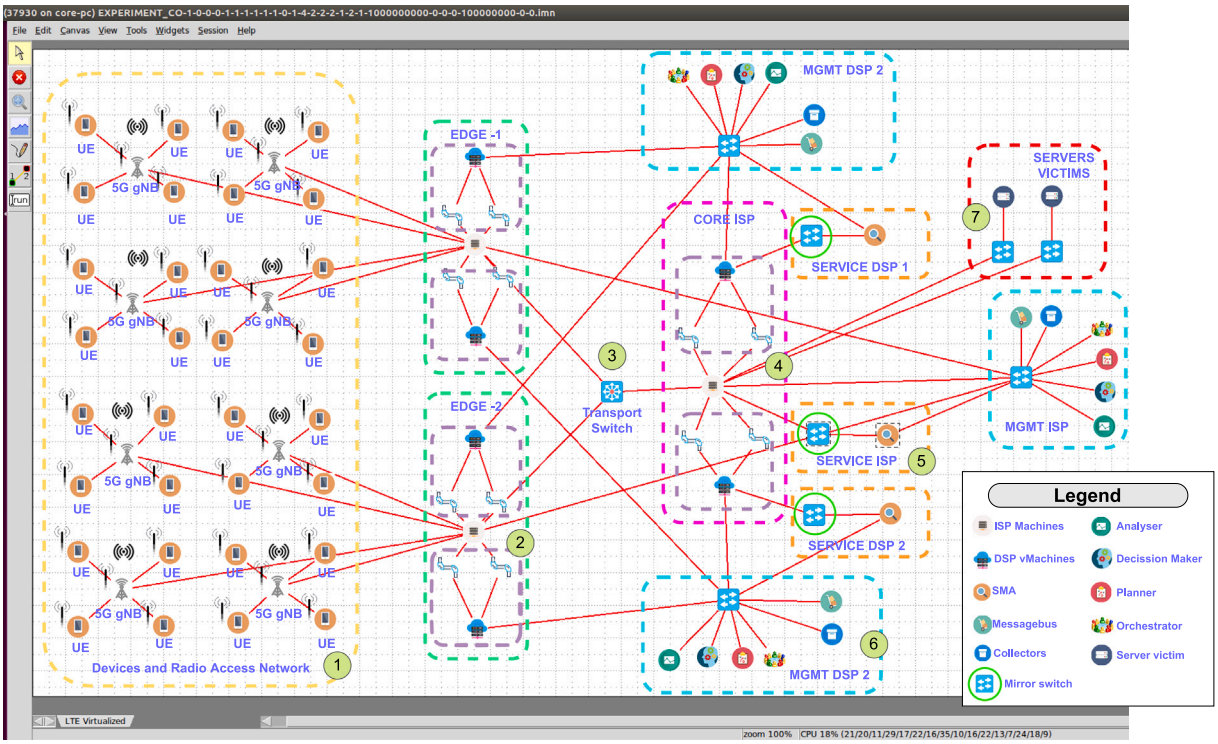

Fig. 5. 32 IoT infected devices in a multi-stakeholder collaborative self protection system over CORE GUI.

in terms of execution times, this being represented by the green area which narrows until it intersects with the following blue area between 16 and 32 devices, a point where, although the average time value for the standalone system is slightly higher than the collaborative one, it is shown that its growth trend is already much more accentuated than its counterpart. From this point on wards, it can be verified that the growth in execution times for the standalone system follows linearly the growth in the number of attackers. On the other hand, the collaborative system presents a high load balancing between the different stakeholders of the system, so that its growth in line with the number of attackers is not as sharp as that of the standalone system. It should be noted that, the use of the proposed system for a small group of attackers might not be worthwhile as a single security solution would be able to cope with the whole attack. However, it is also worth mentioning that there is a continuously growing large number of additional devices being connected to the network considering the trends such as Internet of Things and Bring Your Own Devices, and thus it makes sense to assume that the number of devices will be even higher in the future to justify the purpose and effectiveness of the proposed system. In addition, it is worth to mention that for the more severe attack, there is an improvement of 316% in the execution time of the collaborative system with respect to the standalone own. Each of the attacks is sending 10,000 packets per second (pps) of 160 Maximum Transmission Unit (MTU), totalling 2.56 Mpps.

The second indicator is the ability of each of the systems to detect and stop attacks. This is represented in Fig. 7 below, where, again, a series of bars represents the results for each of the systems. The number of attacks that the system should have stopped has been represented with yellow coloured areas so that the performance of each of the systems is more clearly reflected. In addition, the contribution of each of the stakeholders to the collaborative system has been represented in different shades of blue. As in the previous Fig. 6, the green bars represent the standalone system. It can be seen how, as in the previous figure, when going from 16 to 32 devices there is a saturation in the standalone system that causes its first significant increase in time and the beginning of the decrease in the number of attacks stopped. This decrease will continue for the remaining number of attackers, demonstrating a trend of system saturation that is solved. It is worth to mention that for 128 simultaneous attackers, the collaborative system is able to stop 100% of the attackers while the standalone counterpart is only able to stop a 15% of them, clearly demonstrating the benefits of this collaboration.

Fig. 8 shows the performance of each of the systems in a proportional way. The green line represents the results of the standalone system, while the blue line shows the values obtained for the collaborative system. A blue area has been filled in to observe the drastic difference in performance offered by each of the systems, and to ensure that the load balancing provided by the collaborative system to the network infrastructure can be beneficial for the stakeholders involved in it. It can be seen that for the highest number of attackers (256 attackers) the collaborative system has barely dropped to 78.12% performance, while the standalone system is providing 4.75% performance.It is worth to mention that this reduction of efficiency is also due to the fact that the whole experiment involving management, control and data plane is being executed in the same physical same machine and thus the components of the loops are also indirectly affected by the attack which is in our humble opinion a way to show as well the resilience of the architecture proposed.

There are two concrete scenarios that are worth studying more in deep. The first one is the last one that achieved the efficiency of 100% of attack mitigation on the collaborative system (i.e. 128 attackers) and the second one is the one with highest number of attackers showing some saturation for both systems, i.e. (256 attackers).

Figs. 9 and 10, both representative of the results for a number of 128 attackers. Both graphs are plotted on the contribution of each of the stakeholders involved and, separated by a vertical dashed red line, both systems, the standalone (represented as ISP-ST) on the left of each graph against the collaborative system on the right of each graph. In addition, it has been considered of special interest the representation of the times of each of the components involved in the loop, as follows: the blue coloured area represents the Analyser time; secondly, the red coloured areas represent the average Decision Maker time; the yellow

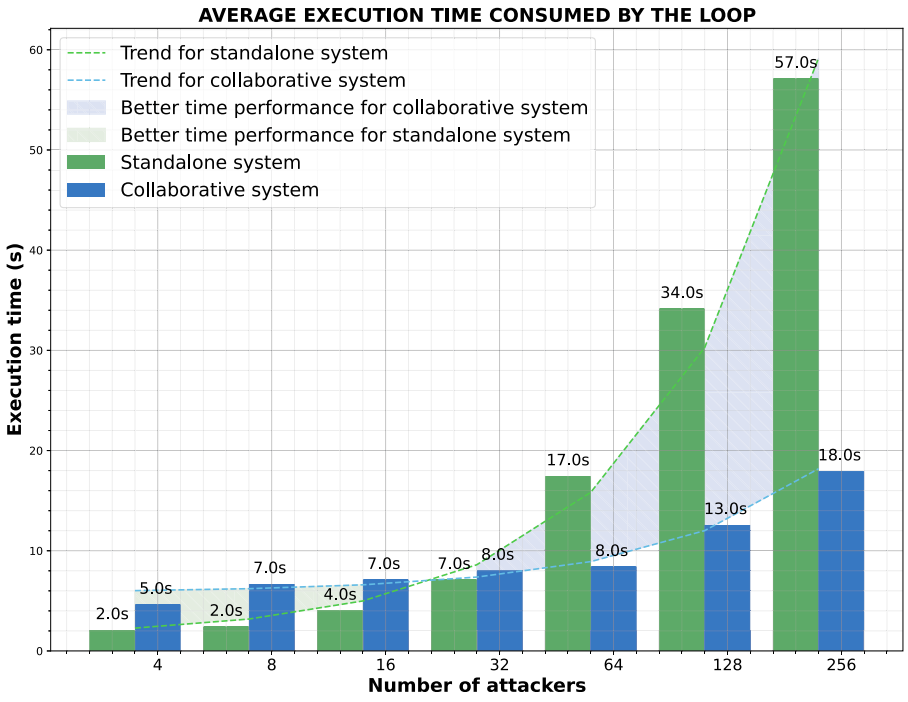

Fig. 6. Average time consumed by the self protection loop, standalone system (green) against the collaborative system (blue). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

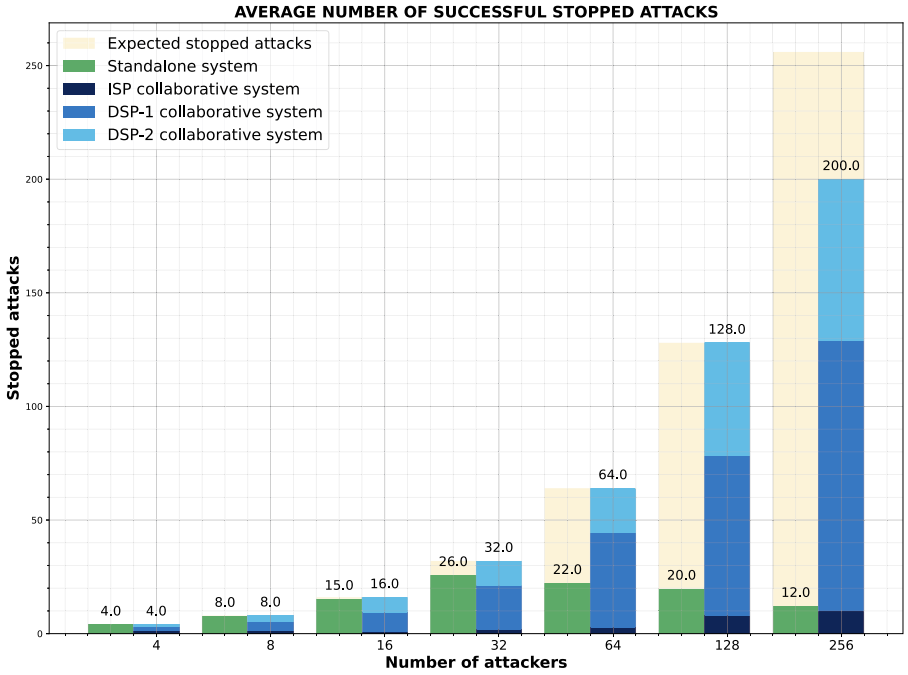

Fig. 7. Average number of stopped attacks by the self protection loop, standalone system (green) against the collaborative system (blue). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

areas represent the average execution time of the Planner component; finally, the green areas represent the Mitigation time, which is the time consumed by the Orchestrator component and the addition of the rule by the FCA component to the point of the network suitable for stopping the attack that has just been analysed.

In Fig. 9, a first significant change is the average time consumed by the Planner component. In the standalone system it shows a slightly lower performance, due to the saturation of resources at that time – it is recalled that in Fig. 7, the standalone system showed a rather steep trend as the number of attackers increased, and 128 was a number where high saturation was shown – while, on the other hand, the Planner component for the collaborative system shows significantly lower times. This is due to the load balancing of the system resources, which allows the loop to act more smoothly and efficiently in stressful situations such as a DDoS attack. On the other hand, there is also a significant change in the Intention times. While they are somewhat lower for the standalone system, the collaborative system shows a certain degree of saturation. It is undoubtedly the component that

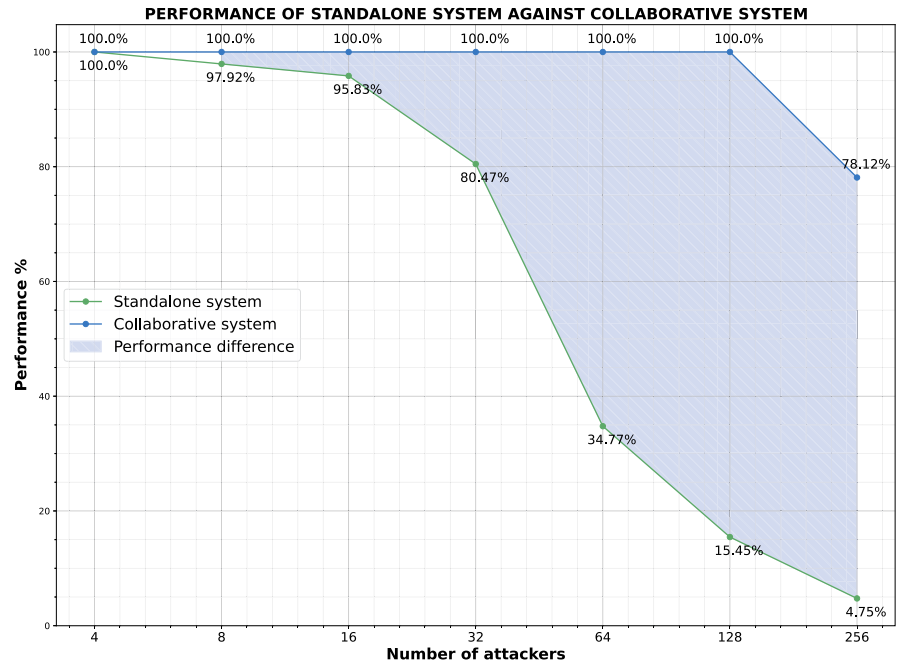

Fig. 8. Performance by the self protection loop, standalone system (green) against the collaborative system (blue). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

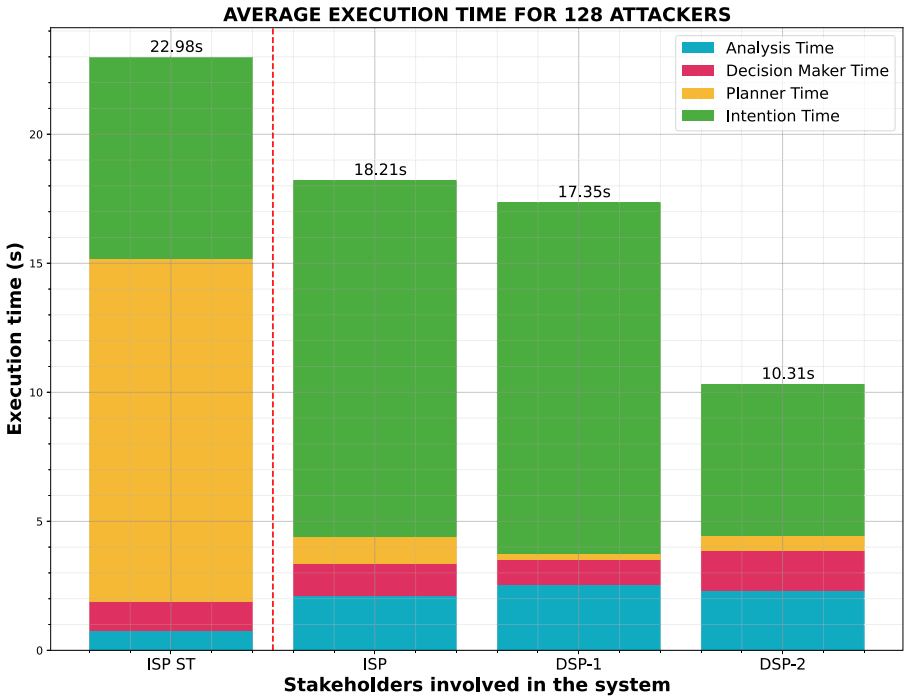

Fig. 9. Experiment results for a number of 128 attackers as the average time consumed by the loop per component. Standalone system is represented as ISP-ST (left) and ISP, DSP-1 and DSP-2 represent the collaborative system (right).

contributes the most delay to the system. However, the collaborative system still maintains average times below the standalone system. Fig. 10 shows the contribution of each of the stakeholders to the number of stopped attacks. It shows how for 128 attackers, the standalone system barely manages to stop 25 attacks on average (19.78%), while the collaborative system shows a load balancing that allows it to stop up to 128 (100%) attackers. It is very relevant to see how the vast majority of the cyber attacks are stopped by the DSP, thus leaving the percentage of resources consumed by the ISP at minimum (8%) even with 100% of the attackers have been stopped in the collaborative system. This result clearly show how each system is benefiting of a mutual collaboration against the same cyber attack even without any information exchanged between them.

The same graphs have been plotted in Figs. 11 and 12 for a number of 256 attackers. In Fig. 11, the same trend can be seen: the standalone system is saturated, causing the execution times of the self-protection loop to be drastically increased, while, on the other hand, the collaborative system shows a great load balancing between each of its stakeholders, achieving more reasonable execution times of the self-protection loop than its counterpart. The same is depicted in Fig. 12,

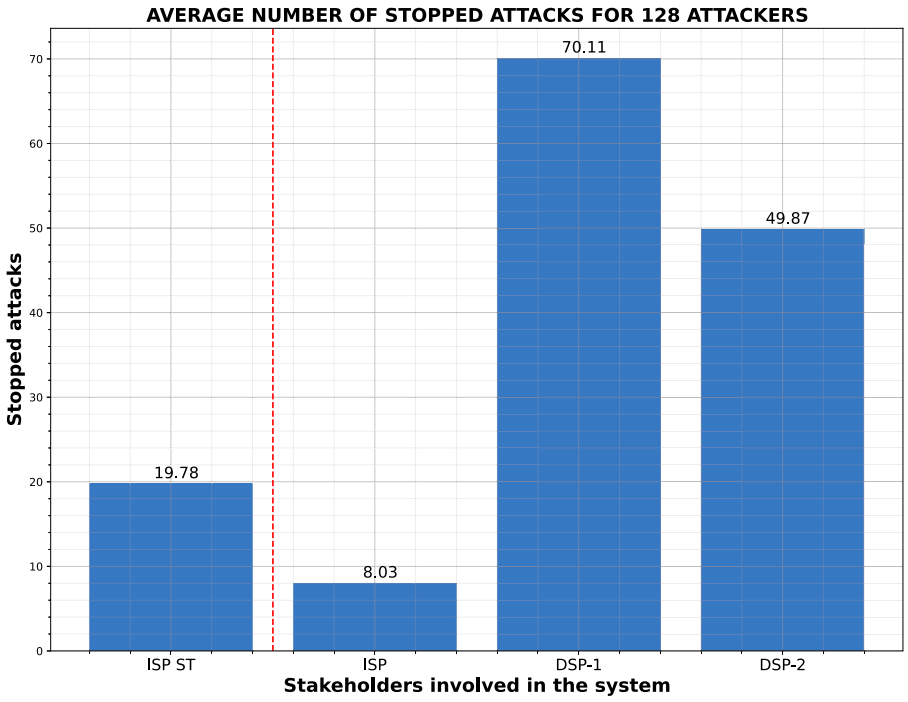

Fig. 10. Experiment results for a number of 128 attackers as the average number of attacks stopped by each stakeholder. Standalone system is represented as ISP-ST (left) and ISP, DSP-1 and DSP-2 represent the collaborative system (right).

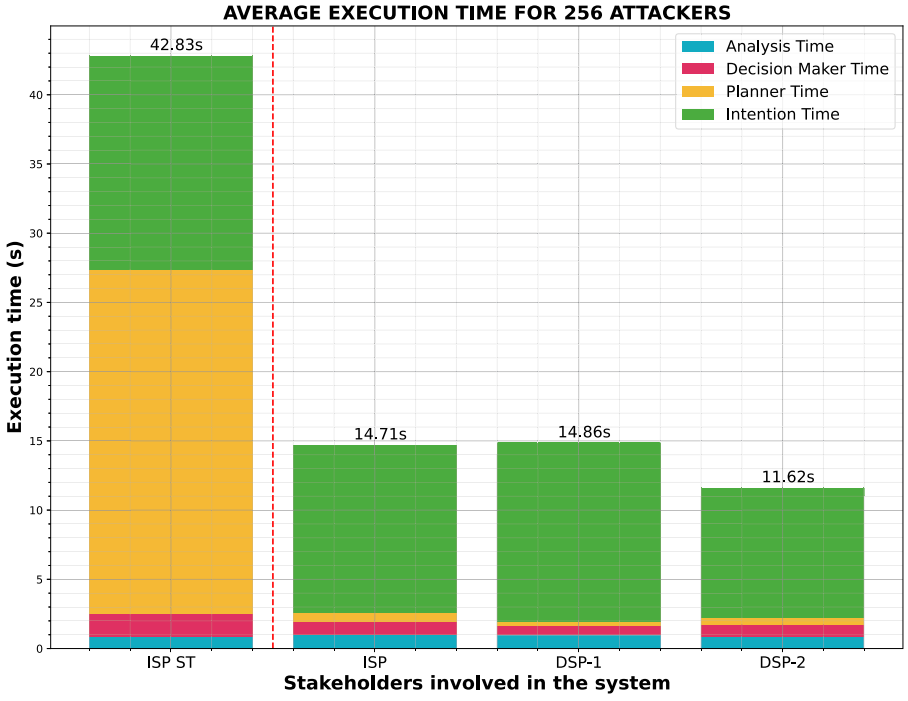

Fig. 11. Experiment results for a number of 256 attackers as the average time consumed by the loop per component. Standalone system is represented as ISP-ST (left) and ISP, DSP-1 and DSP-2 represent the collaborative system (right).

which shows the number of attacks stopped by each of the systems, with the standalone system being really affected by the saturation of the network itself, as opposed to the load balancing offered by the collaborative system, which stops an average of 200 attackers. Notice how the collaborative system has increased the percentage of efficiency from 4.75% to 72.18%, clearly demonstrating the effectiveness of the proposed architecture.

6. Conclusion

Throughout this work the novelty of a collaborative self-protection system between different stakeholders of a 5G/6G network has been studied, supporting both Edge and Core networks in order to detect, analyse and orchestrate the mitigation of a massive DDoS attack. This

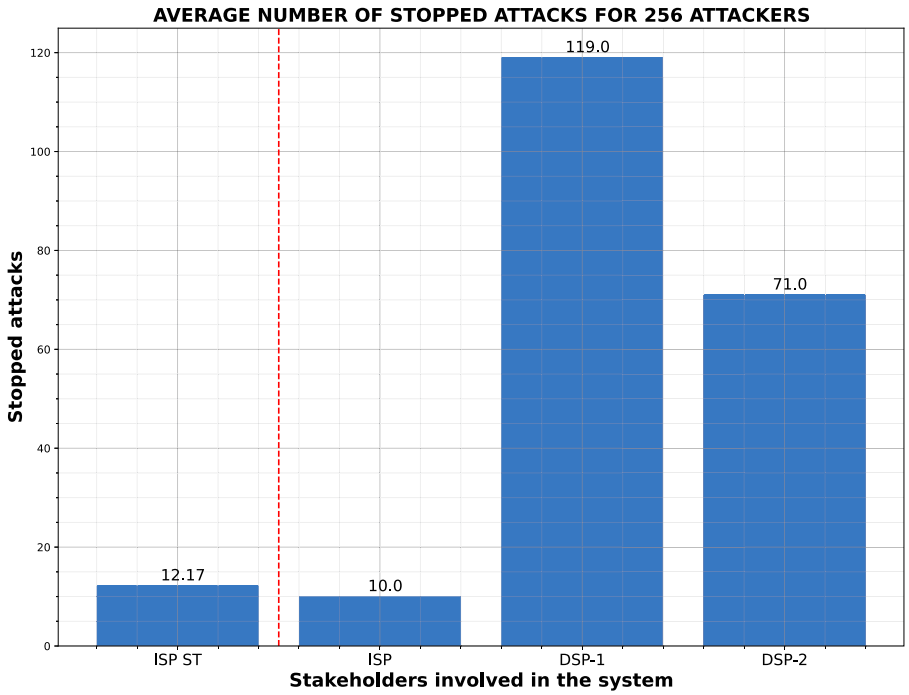

Fig. 12. Experiment results for a number of 256 attackers as the average number of attacks stopped by each stakeholder. Standalone system is represented as ISP-ST (left) and ISP, DSP-1 and DSP-2 represent the collaborative system (right).

approach manages to provide protection and security to the stakeholders involved in the network, with the benefit of not exchanging information with each other, working collaboratively yet autonomously at the same time. It has also been possible to test the effectiveness of the self-protection loop composed by a NIDS and followed by the subsequent components: SMA, Analyser, Decision Maker, Planner, Orchestrator and FCA, with which the detection, analysis and coordination of the mitigation for the protection of its network resources is conducted.

It has been observed that the same system is equally effective for different network topologies that may occur dynamically, and can be sufficiently intelligent and reactive to the change and addition of new infected IoT devices to maintain a good rate of detection and mitigation of attacks. The results of the experiments have shown that the main contribution of this paper (which consists of spreading the work of the protection layers among the different stakeholders involved in the system as mentioned in Section 1), achieves a better performance both in execution time and the number of attacks mitigated compared with the traditional approach where the infrastructure provider has always been the only one concerned with protecting its networks and services, whereas, as has been demonstrated, other stakeholders are also interested in keeping their digital services available and secured. Based on industry surveys, network downtime reaches up to $5,600 per minute of capital costs, which is over $300K per hour [44]. In addition, 36% of companies experiencing more than five DDoS attacks suffer an average downtime of seven to 12 h [45]. This contribution has empirically demonstrated that the proposed solution is able to stop and recover the resources from a DDoS attack involving a large number of attackers with a low response time that is less than a minute. The contribution can thus directly reduce the average capital and operational costs from $3M to less than $1,900 per incident, as the highest response time for this contribution is 20 s, 1/3 of a minute.

The advantage of the simultaneous execution of the loops in different stakeholders is a naturally created multi-layer protection with enhanced distributed capabilities against cyber-attacks and this improves the usage of distributed resources in different stakeholders' domains for advanced protection purposes. Consequently, the performance of the whole protection capabilities is significantly strengthened.

Furthermore, the architecture of the self-protection loop developed in this work is completely modular and extensible. New functionalities can be added so as to optimize the levels of protection and security within the system to be protected, depending on the new use case to be studied. The number of rules and mitigation strategies to be followed can also be further expanded depending on the type of attack or threat.

In future work, firstly, it has been concluded that resource limits are considered a major bottleneck when it comes to replicating the scenarios and attacks carried out. This is why, as future work, the use of one of the functionalities of the CORE emulator that allows the creation of distributed environments on different machines is proposed, being able to completely isolate the management layers from the rest of the infrastructures, thus not being affected by the consumption of resources by the infected devices on the same machine. Another point of interest to be studied is the possible addition of a new component to the self-protection loop endowed with AI and capable of a more precise analysis of the attack that may be occurring on the network and that could replace the NIDS based on checking rules. In addition, it is planted to continue improving the performance for the collaborative system in terms of the execution times of the loop and the number of stopped attacks for even larger DDoS attacks, with the ambition to able to stop the 100% of the attacks launched in the ranges of 4096 to 16,384 infected devices.

CRediT authorship contribution statement

Pablo Benlloch-Caballero: Conceptualization, Software, Validation, Writing – original draft. **Qi Wang:** Writing – review & editing, Supervision. **Jose M. Alcaraz Calero:** Conceptualization, Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: PABLO BENLLOCH-CABALLERO reports financial support was provided

by EU Framework Programme for Research and Innovation ICT Leadership in Enabling and Industrial Technologies. PABLO BENLLOCH-CABALLERO reports financial support was provided by European Commission. JOSE M. ALCARAZ CALERO reports financial support was provided by European Commission. QI WANG reports financial support was provided by European Commission. "This work is funded in part by the European Commission under Grant Agreements H2020-SU-DS-2018-2019-2020/101020259 ARCADIAN- IoT: Autonomous Trust, Security and Privacy Management Framework for IoT) and H2020-ICT-2020-2/101017226 (6G BRAINS: Bringing Reinforcement learning Into Radio Light Network for Massive Connections)".

Data availability

The data that has been used is confidential.

References

[1] Lionel Sujay Vailshery, Global IoT market size, https://www.statista.com/statistics/976313/global-iot-market-size/.
[2] Nikolay Pankov, Protect networked IoT devices or protect the network from IoT devices? https://www.kaspersky.com/blog/rsa2021-dangerous-iot/40161/.
[3] esentire, Reaper IoT Botnet, https://www.esentire.com/security-advisories/reaper-iot-botnet.
[4] McKinsey&Company, The road to 5G: The inevitable growth of infrastructure cost, https://www.mckinsey.com/industries/technology-media-and-telecommunications/our-insights/the-road-to-5g-the-inevitable-growth-of-infrastructure-cost.
[5] Mark Jackson, VoIP Provider Voipfone UK knocked out by DDoS attack AGAIN UPDATE, https://www.ispreview.co.uk/index.php/2021/10/voip-provider-voipfone-uk-knocked-out-by-ddos-attack-again.html.
[6] DDoS attack trends for 2022 Q2, Clodfare Blog, https://blog.cloudflare.com/ddos-attack-trends-for-2022-q2/.
[7] 3GPP, Study on CU-DU Lower Layer Split for NR, Technical Report V15.0.0, 3GPP, 2017.
[8] 3GPP, LTE; 5G; Evolved Universal Terrestrial Radio Access (E-UTRA) and NR; Service Data Adaptation Protocol (SDAP) specification (3GPP TS 37.324 version 16.2.0 Release 16), Technical Report V16.2.0, ETSI, 2020.
[9] 3GPP, Radio Resource Control (RRC); Protocol specification, Technical Report V16.8.0, 3GPP, 2022.
[10] 3GPP, Packet Data Convergence Protocol (PDCP); Protocol specification, Technical Report V16.6.0, 3GPP, 2021.
[11] Mobile Edge Computing a Key Technology Towards 5G, Technical Report, ETSI, 2015.
[12] Multi-access Edge Computing (MEC); Phase 2: Use Cases and Requirements, Technical Report, ETSI, 2018.
[13] Multi-access Edge Computing (MEC); Framework and Reference Architecture, Technical Report, ETSI, 2020.
[14] Christian Mannweiler Simone Redana, View on 5G Architecture, Technical Report, 5G PPP, 2020.
[15] Christos Tzagkarakis, Nikolaos Petroulakis, Sotiris Ioannidis, Botnet attack detection at the IoT edge based on sparse representation, in: 2019 Global IoT Summit, GIoTS, 2019, pp. 1–6, http://dx.doi.org/10.1109/GIOTS.2019.8766388.
[16] Tolijan Trajanovski, Ning Zhang, An automated and comprehensive framework for IoT botnet detection and analysis (IoT-BDA), IEEE Access 9 (2021) 124360–124383, http://dx.doi.org/10.1109/ACCESS.2021.3110188.
[17] Mikail Mohammed Salim, Sushil Kumar Singh, Jong Hyuk Park, Securing smart cities using LSTM algorithm and lightweight containers against botnet attacks, Appl. Soft Comput. 113 (2021) 107859, http://dx.doi.org/10.1016/j.asoc.2021.107859.
[18] Ying Liu, Ting Zhi, Ming Shen, Lu Wang, Yikun Li, Ming Wan, Software-defined DDoS detection with information entropy analysis and optimized deep learning, Future Gener. Comput. Syst. 129 (2022) 99–114, http://dx.doi.org/10.1016/j.future.2021.11.009.
[19] Bilal Hussain, Qinghe Du, Bo Sun, Zhiqiang Han, Deep learning-based DDoS-attack detection for cyber–physical system over 5G network, IEEE Trans. Ind. Inform. 17 (2) (2021) 860–870, http://dx.doi.org/10.1109/TII.2020.2974520.
[20] Zubair A. Baig, Surasak Sanguanpong, Syed Naeem Firdous, Van Nhan Vo, Tri Gia Nguyen, Chakchai So-In, Averaged dependence estimators for DoS attack detection in IoT networks, Future Gener. Comput. Syst. 102 (2020) 198–209, http://dx.doi.org/10.1016/j.future.2019.08.007.
[21] Renato S. Silva, Carlos Colman Meixner, Rafael S. Guimarães, Thierno Diallo, Borja O. Garcia, Luís F.M. de Moraes, Magnos Martinello, REPEL: A strategic approach for defending 5G control plane from DDoS signalling attacks, IEEE Trans. Netw. Serv. Manag. 18 (3) (2021) 3231–3243, http://dx.doi.org/10.1109/TNSM.2020.3035342.
[22] Zhuotao Liu, Yuan Cao, Min Zhu, Wei Ge, Umbrella: Enabling ISPs to offer readily deployable and privacy-preserving DDoS prevention services, IEEE Trans. Inf. Forensics Secur. 14 (4) (2019) 1098–1108, http://dx.doi.org/10.1109/TIFS.2018.2870828.
[23] Jiaqi Li, Zhifeng Zhao, Rongpeng Li, Machine learning-based IDS for software-defined 5G network, IET Netw. 7 (2) (2018) 53–60, http://dx.doi.org/10.1049/iet-net.2017.0212.
[24] David Candal-Ventureira, Pablo Fondo-Ferreiro, Felipe Gil-Castiñeira, Francisco Castaño, Quarantining malicious IoT devices in intelligent sliced mobile networks, Sensors (Basel, Switzerland) 20 (2020) http://dx.doi.org/10.3390/s20185054.
[25] Paloalto Networks, 5G Network Slice Security, https://docs.paloaltonetworks.com/service-providers/10-0/mobile-network-infrastructure-getting-started/5g-security/5g-network-slice-security.html.
[26] Ana Serrano Mamolar, Pablo Salvá-García, Enrique Chirivella-Perez, Zeeshan Pervez, Jose M. Alcaraz Calero, Qi Wang, Autonomic protection of multi-tenant 5G mobile networks against UDP flooding DDoS attacks, J. Netw. Comput. Appl. 145 (2019) 102416, http://dx.doi.org/10.1016/j.jnca.2019.102416.
[27] Cisco, VXLAN overview: Cisco Nexus 9000 series switches, 2013, Cisco White Papers, URL https://www.cisco.com/c/en/us/products/collateral/switches/nexus-9000-series-switches/white-paper-c11-729383.html.
[28] Enrique Chirivella-Perez, Juan Gutiérrez-Aguado, Jose M. Claver, Jose M. Alcaraz Calero, Hybrid and extensible architecture for cloud infrastructure deployment, in: 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing, 2015, pp. 611–617, http://dx.doi.org/10.1109/CIT/IUCC/DASC/PICOM.2015.87.
[29] Enrique Chirivella-Perez, Juan Gutiérrez-Aguado, Jose M. Claver, Jose M. Alcaraz Calero, Hybrid and extensible architecture for cloud infrastructure deployment, in: 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing, 2015, pp. 611–617, http://dx.doi.org/10.1109/CIT/IUCC/DASC/PICOM.2015.87.
[30] Apache Kafka, https://kafka.apache.org/24/documentation.html,
[31] RabbitMQ, RabbitMQ, https://www.rabbitmq.com/documentation.html.
[32] Ignacio Sanchez-Navarro, Ana Serrano Mamolar, Qi Wang, Jose M. Alcaraz Calero, 5GTopoNet: Real-time topology discovery and management on 5G multi-tenant networks, Future Gener. Comput. Syst. 114 (2021) 435–447, http://dx.doi.org/10.1016/j.future.2020.08.025.
[33] Ignacio Sanchez-Navarro, Jorge Bernal Bernabe, Jose M Alcaraz-Calero, Qi Wang, Advanced spatial network metrics for cognitive management of 5G networks, Soft Comput. 25 (1) (2021) 215–232, http://dx.doi.org/10.1007/s00500-020-05132-y.
[34] Snort Intrusion Detection System (IDS), Snort, https://www.snort.org/.
[35] Ana Serrano Mamolar, Zeeshan Pervez, Jose M. Alcaraz Calero, Asad Masood Khattak, Towards the transversal detection of DDoS network attacks in 5G multi-tenant overlay networks, Comput. Secur. 79 (2018) 132–147, http://dx.doi.org/10.1016/j.cose.2018.07.017.
[36] Antonio Matencio Escolar, Jose M. Alcaraz-Calero, Pablo Salva-Garcia, Jorge Bernal Bernabe, Qi Wang, Adaptive network slicing in multi-tenant 5G IoT networks, IEEE Access 9 (2021) 14048–14069, http://dx.doi.org/10.1109/ACCESS.2021.3051940.
[37] Pablo Salva-Garcia, Jose M. Alcaraz-Calero, Qi Wang, Jorge Bernal Bernabe, Antonio Skarmeta, 5G NB-IoT: Efficient network traffic filtering for multitenant IoT cellular networks, Secur. Commun. Netw. 2018 (9291506) (2018) http://dx.doi.org/10.1155/2018/9291506.
[38] Pablo Salva-Garcia, Jose M. Alcaraz-Calero, Qi Wang, Miguel Arevalillo-Herráez, Jorge Bernal Bernabe, Scalable virtual network video-optimizer for adaptive real-time video transmission in 5G networks, IEEE Trans. Netw. Serv. Manag. 17 (2) (2020) 1068–1081, http://dx.doi.org/10.1109/TNSM.2020.2978975.
[39] Coreemu, CORE: Common open research emulator, 2022, GitHub, https://github.com/coreemu/core.
[40] Open vSwitch, Open VSwitch, https://docs.openvswitch.org/en/latest/index.html.
[41] TUN/TAP Device Driver, The Linux Kernel, https://www.kernel.org/doc/html/latest/networking/tuntap.html.
[42] OsmoGGSN, OsmoGGSN, https://osmocom.org/projects/openggsn/wiki.
[43] Markus Goldstein, DDoS botnet simulator (BoNeSi), 2018, GitHub, https://github.com/markus-go/bonesi.
[44] Andrew Lerner (Gartner), The cost of downtime, https://blogs.gartner.com/andrew-lerner/2014/07/16/the-cost-of-downtime/.
[45] Ahmad Nassiri, This is how much time and money a DDoS attack will cost you, https://www.a10networks.com/blog/this-is-how-much-time-and-money-ddos-attack-will-cost-you/.

Pablo Benlloch-Caballero is a Ph.D. candidate at School of Engineering and Computing at the University of West of Scotland, where he is actively involved with his research project about autonomous cybersecurity in industrial IoT environments. Nowadays he is also working in the H2020-SU-DS-2018-2019-2020/101020259 ARCADIAN-IoT: Autonomous Trust, Security and Privacy Management Framework for IoT). Some of his main research interests are: artificial intelligence applied to cybersecurity, Network security and 5G/6G Networks in cloud computing. He has a B.Sc. Honours in Telecommunications Engineering degree at Universitat de València and a Master's Engineering Telecommunications at Universitat Politècnica de València, both at Spain.

Prof Qi Wang (M) is a Professor in Next-Generation Networks with AVCN at UWS. He is a winner of 2018 Scottish Knowledge Exchange Award, 2018 NATO Scientific Achievement Award, and 2020 Scotland CeeD Industry Awards – Innovation Award. He is a Board Member of the EU 5G-PPP Technology Board, Member of several 5G-PPP Working Groups, and Scotland's Developing AI and AI Enabled Products and Services Working Group. He is the CoTechnical Manager for EU Horizon 2020 projects SELFNET and SliceNet, and Principal Investigator or Co-Investigator for numerous projects sponsored by various funding bodies including UK EPSRC, UK Ministry of Defence, InnovateUK, Knowledge Transfer Partnership, Innovation Centre for Sensor and Imaging Systems and Royal Society of Edinburgh. He has published more than 150 papers in these areas in leading international journals such as various IEEE Transactions (on Broadcasting, Services Computing, or Vehicular Technologies etc.), conferences or books. He was a Best Paper Award Winner of IEEE ICCE 2014, IEEE ICCE2012, SOFTNETWORKING 2017, and SIGMAP 2014.

Prof Jose M. Alcaraz Calero (M) is a Professor in Networks at School of Engineering and Computing at the UWS. He is an IEEE Senior Member with experience in 5G networks, network slicing, monitoring, automation and management. In the academic side, he has more than 150 publications in SCI-indexed international journals, conferences and books. He has been involved in 20 editorial activities in the most prestigious journal in the field and served as chair in 20 international flagship conferences and contributes as Technical Programme Committee member in more than 100 international conferences. In the industrial side, he has more than 50 patents and intellectual property rights. From the leadership perspective, Jose M has significant experience as Principal Investigator or as Co-Investigator in more than 20 research projects at local, national and especially at European and international level. He is the CoTechnical Manager for EU Horizon 2020 projects SELFNET and SliceNet.