Security-as-a-Function for IDS/IPS in Softwarized Network and Applications to 5G Network Systems

Shivank Malik and Samaresh Bera, *Senior Member, IEEE*

***Abstract*—The service-based architecture of 5G network allows network operators to place virtualized network functions on commodity hardware, unlike the traditional vendor-specific hardware-based functionalities. However, it expands the security vulnerabilities/threats to the 5G network. While there exist several theoretical studies on network function placement and service routing, a few focused on the security aspects of the 5G network systems.**

This paper focuses on safeguarding the 5G core network systems from DoS/DDoS attacks by placing intrusion detection/prevention systems (IDS/IPS) as virtualized network functions following the 5G standalone architecture. To ensure the virtualized placement of IDS/IPS, first, we provide thorough virtual machine (VM)-based and containerized implementation details and evaluate the network performance with two scenarios – IDS and IPS – in the presence of TCP and UDP applications. Second, we apply the VM-based implementation of IDS/IPS on a softwarized 5G core network and study the network performances. The experiment results on network throughput, latency, and packet drop reveal that the softwarized IDS/IPS can meet the QoS requirements of 5G applications, while safeguarding the network from DoS/DDoS attacks.

***Index Terms*—Softwarized security functions, Quality-of-service (QoS), Intrusion detection system (IDS), Intrusion prevention system (IPS), Network performance, 5G network security**

I. Introduction

The recent advancement of mobile communications (such as 5G and beyond networks) and industry 5.0 is expected to support modern applications with stringent quality-of-service (QoS) requirements. The 5G applications are categorized into three broad areas – enhanced mobile broadband (eMBB), ultra-reliable and low-latency communications (uRLLC), and massive machine-type communications (mMTC) [2], [3]. The QoS requirements of these applications range from high bandwidth to high reliability to low latency [4]. Whereas Industry 5.0 requires network operators to enable smart manufacturing with improved resilience and sustainability [5]. Furthermore, it also requires the network operators to support a massive number of smart devices (such as robots) in a small and congested deployment area. Consequently, heterogeneous applications need to be supported by the network while preserving security to the network elements. The ultimate choice is to place closed-box vendor-specific hardware devices to ensure a secure system. However, fulfilling such diverse QoS requirements of the applications using traditional vendor-specific networking architecture and devices are complex and cost-expensive [6].

The softwarized network supported by software-defined networking (SDN) and network function virtualization (NFV) technologies addresses the issues with vendor-specific networking devices [7]–[9]. The SDN provides flexible networking for traffic forwarding by separating the control and data planes. Whereas NFV enables network operators to place virtual network functions (VNFs) on commodity hardware as per requirements. Therefore, with the help of SDN and NFV, customized network functions can be placed as VNFs on commodity hardware in the form of virtual machines (VMs) or containers [8], [10], [11].

The service-based architecture of 5G network network also allows the use of virtualized network functions instead of closed-box hardware devices [12]. Furthermore, there exist significant studies on the placement of VNFs, traffic scheduling through the VNFs, and efficient resource utilization at the commodity hardware [13]–[17]. However, these studies focused on theoretical modeling of the problem and did not focus on the practical implementation and their impact on network performance, i.e., whether the QoS requirements of the underlying applications can be fulfilled using softwarized functions. Moreover, the use of softwarized network expands the security threats/vulnerabilities compared to the traditional hardware-based networking. Furthermore, the network performance may be impacted with softwarized functions compared to the hardware-based systems. While there exist a few works on studying the impact on network performances [18]–[23], a comprehensive study is required considering emerging applications with stringent QoS requirements, as presented in Table I [4], to analyze the feasibility of using softwarized network functions in 5G network system while ensuring security.

In this paper, we study the implementation and impact of softwarized network security functions on network performance in a softwarized 5G network. In particular, we focus on the primary network security functions, such as intrusion detection system (IDS) and intrusion prevention system (IPS). We note that we utilize both the containerized and VM-based implementations of the security functions on commodity hardware. The objective of performing this study in both VM and containerized environment is to obtain and analyze a holistic overview of the security functions on network performance and not to compare them with physical network functions. Next, we apply the softwarized IDS/IPS functions in a softwarized 5G network to safeguard the other network functions in the 5G core network from DoS/DDoS attacks. We also study the network performance in the 5G network in terms of throughput, latency, and packet drop. The key contributions

S. Malik and S. Bera are with the Department of Computer Science and Engineering, Indian Institute of Technology Jammu, Jammu and Kashmir, India, 181221. Email: {2022pis0074@iitjammu.ac.in, s.bera.1989@ieee.org}.

A preliminary version [1] of this work has been accepted for publication in the International Conference on Signal Processing and Communications (SPCOM), 2024, Bangalore, India.

TABLE I: QoS parameters of different applications [4]

Scenario		Latency	Data rate	Reliability	Payload
Discrete automation		10 ms	10 Mbps	99.99%	small to high
Process automation	remote control	60 ms	1 to 100 Mbps	99.999%	small to high
	monitoring	60 ms	1 Mbps	99.9%	small
Electricity distribution	medium voltage	40 ms	10 Mbps	99.9%	small to high
	high voltage	5 ms	10 Mbps	99.999%	small
Intelligent transportation system	infrastructure backhaul	30 ms	10 Mbps	99.9999%	small to high

TABLE II: Comparison between the existing studies and this work

Work	Security Functions			Performance Analysis				Environment		Use Case
	IDS	IPS	NAT	Thr.	Lat.	Jit.	PD	VM	Containers	
Gallenmuller et al. [18]	✓	✓	✗	✗	✓	✗	✗	✓	✗	✗
Seeber et al. [23]	✓	✗	✗	✗	✗	✗	✗	-	-	✗
Fadhilah and Marzuki [22]	✓	✗	✗	✗	✗	✗	✓	✓	✗	✗
Aggarwal and Thangaraju [24]	✗	✗	✗	✓	✗	✗	✗	✓	✓	✗
Li et al. [25]	✗	✗	✗	✓	✗	✗	✗	✓	✓	✗
Garg et al. [26]	✓	✗	✗	✗	✗	✗	✗	✗	✓	✗
Gupta and Sharma [15]	✓	✗	✗	✗	✗	✗	✓	✓	✗	✗
Varghese and Muniyal [19]	✓	✗	✗	✓	✓	✗	✗	✓	✓	✗
This work	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

in this work are as follows:

- We implement a secure softwarized network with IDS and IPS virtual network functions as VMs and containers placed on commodity hardware. Furthermore, we utilize open-source software tools to implement the network security functions.
- We provide comprehensive details of the network setup for two implementation scenarios, `IDS` and `IPS`, for both VM-based and containerized placement.
- We use the D-ITG [27] traffic generator to generate network traffic for TCP and UDP applications with varying QoS requirements. The extensive experiment results on throughput, latency, jitter, and packet drop are analyzed considering diverse QoS requirements of different applications, as presented in Table I.
- We apply the network security functions in a softwarized 5G network and build a prototype of the entire system. Then we present the results on network throughput, latency, and packet drop in the presence of the security functions. The results show that the 5G network can meet the QoS requirements of emerging 5G applications, while safeguarding it from the DoS/DDoS attacks.

The remainder of the paper is organized as follows. Section II presents an overview of the existing studies. Section III describes the detailed network setup and implementation of a secure softwarized network. Section IV presents the experiment results with a detailed analysis. Section V presents the 5G network prototype with security functions and results on network performance. Finally, Section VI concludes the paper with future research directions.

II. Background

In this section, we discuss the exiting studies on IDS/IPS and their impacts on network performance [4], [10], [15]–[23], [25], [28].

Fadhilah and Marzuki [22] discussed the impact of hardware configuration of VMs using different numbers of CPU cores while creating different intrusion-based attack scenarios in the network. In another study, Gallenmuller et al. [18] evaluated the impact of Snort-based forwarding and filtering strategies on latency performance. The authors evaluated the performance of the Snort-based security functions in two scenarios – functions placed on the host machine and VMs. The results show that VM-based security functions impose increased latency compared to the direct placement on the host machine.

Shams et al. [20] studied the comparison of different security functions in a software-defined networking platform. The authors integrated an open-source IDS software with SDN and tested their capabilities against malicious traffic generated by an attacker. Varghese and Muniyal [19] proposed a framework for distributed denial-of-service (DDoS) attacks in SDN environment. The attack report captured at the data-plane of the SDN is sent to the control-plane for making fine-grained traffic forwarding decisions.

Gedia and Perigo [10] presented SDN controller's performance when implemented inside a VM and a containerized platform. In another study, Li et al. [24] analyzed the performance of various virtualization technologies for VNF deployments, highlighting trade-offs between isolation and agility. These studies majorly delve into the performance implications of NFVs. Similarly, the works in [15]–[17] explored the placement and resource utilization of VNFs, highlighting their potential for efficient service provisioning.

The authors in [25], [29] presented a comparison between VM and container overhead, emphasizing the need for careful tuning, especially for I/O-intensive security functions such as IDS and IPS. Beyond direct performance comparisons, Garg et al. [26] examined the challenges associated with migrating VM-based workloads (e.g., security functions) to containers. The study considers resource sharing, concurrency, isolation and dependability, while providing valuable insights into potential performance implications of such migrations.

While some approaches offer potential benefits, a careful consideration of the platform, configuration of security functions and its characteristics are crucial to ensure QoS-aware service provisioning in practical deployments. Furthermore, while a few are aligned with the studied scenarios in this paper, they significantly differ from this work. The key differences are highlighted briefly in Table II and discussed as follows. This work focuses on a comprehensive implementation and performance evaluation of softwarized security functions. While most of the existing works focus on the detection and prevention of intrusions, our objective is to study the impact of softwarized network security functions on network performance. Therefore, we do not focus on the decision making policies on traffic forwarding, i.e., whether to forward or drop or generate alert message upon receiving an incoming network traffic. The evaluation is also done in a softwarized 5G network as a use case scenario in contrast to the existing works.

III. Implementation Details

We follow the ETSI's NFV architecture [30] to deploy and the manage the VNFs in a virtualized environment. Consequently, we use the virtualized infrastructure of a host machine to place the security functions with the following hardware and software configuration: Processor: Intel i9-13th generation, 24 core; RAM: 64GB; Ethernet: 1 Gbps. Figure 1 shows the schematic view of the security framework between the client and server. The network traffic between the client and server is passed through the secure virtualized network based on the security configurations. We consider the following network security functions – IDS and IPS, as shown in Figure 1. We note that the network address translation (NAT) module is the integral part of the network system.

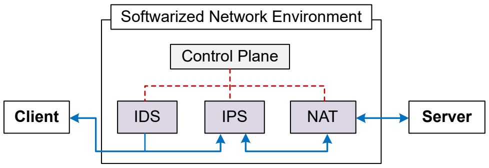
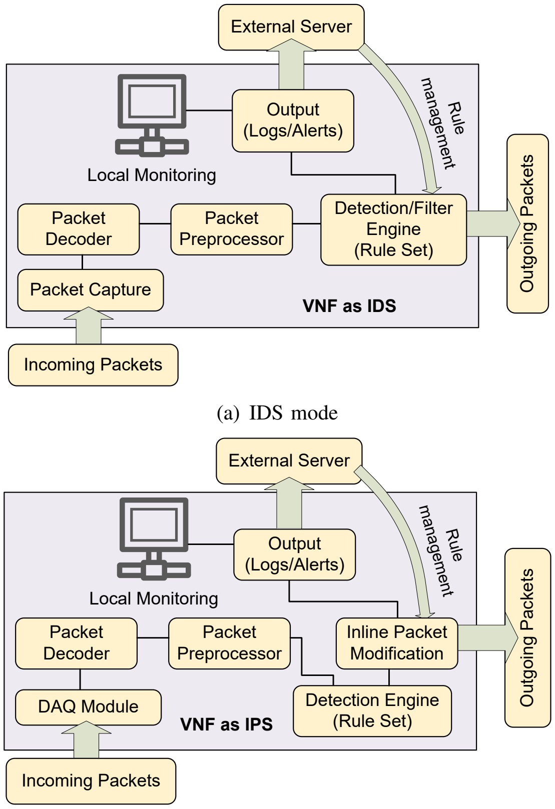


Fig. 1: Schematic view of the framework considered for softwarized security function

Network Setup: We create three guest machines on the host machine, two for the client and the server, and the other to place the softwarized security functions. Similarly, three containers are created to understand the impact of softwarized security functions in containerized environment. We use Snort (https://www.snort.org/), which is an open-source network security tool, to create the IDS and IPS network functions. In case of IPS, we use the data acquisition library (DAQ) (https://github.com/snort3/libdaq) in order to efficiently intercept, analyze and take rule-based actions on the network traffic. At the same time, we use the `iptables` utility available in Linux to create NAT forwarding functionalities. Figures 2(a) and 2(b) present the modules of the VNF in IDS and IPS modes, respectively. Consequently, we create different security scenarios discussed in the subsequent sections.



(a) IDS mode

(b) IPS mode

Fig. 2: Components of the network security function in IDS and IPS modes

A. Scenario 1: Intrusion Detection System (IDS)

Figure 3 shows the network setup between the client and server with IDS software security functions. The IP configuration of client, server, and virtualized network is shown in Figure 3.

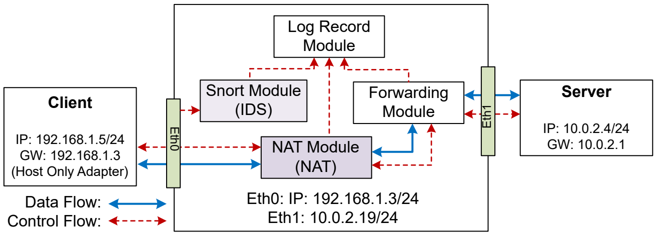


Fig. 3: IDS between client and server

The specific changes made in the network and `iptables` rules are as follows:

```
# enabled ipv4 forwarding
ipv4.forward = 1 # in /etc/sysctl.conf

# to resolve the DNS by the client
sh -c "echo nameserver 8.8.8.8 >\
/etc/resolv.conf"

# for traffic forwarding with NAT
iptables -t nat -A POSTROUTING \
-j MASQUERADE
```

The Snort IDS listens to the Ethernet interface (Eth0) connected to the client and generates alerts based on the rules configured in Snort. The Snort IDS module sends the alert messages to the `log-record module` at the control-plane. We note that the IDS network function silently listens to the interface and generates alert messages based on the forwarding rules configured in snort. It is not involved in making packet forwarding decisions. Therefore, the network performance in IDS scenario is eventually impacted by the NAT network function. We use snort community rules and customized rules for decision making. A sample set of rules used in Snort for IDS is as follows:

```
alert icmp any any -> \
any any (msg:"ADMIN-ALERT, ICMP traffic \
detected";sid:1000004;)

alert tcp any any -> \
$HOME_NET 80 (msg:"Possible HTTP DoS \
Attack";sid:1000002;)

alert icmp any any -> \
$HOME_NET 80 (msg:"Dos Attack suspected";\
sid:1000001;)

alert tcp $EXTERNAL_NET any -> $HOME_NET \
445 (msg: "Exploit Detected!"; \
flow: to_server, established; classtype: \
attempted-admin; priority: 10; \
sid: 2094284; rev: 2;)
```

B. Scenario 2: Intrusion Prevention System (IPS)

Figure 4 shows the second scenario, where the Snort module acts as an IPS. The Snort module utilizes the NFQ to process all packets sent from the client so that the packets pass through the Snort IPS and the desired action is taken – whether to `forward` or `drop` or `send alert`. We note that all packets are allowed through the Snort IPS module in this experiment, as our primary objective is to study the impact on network performance. However, any desired action can easily be integrated into the setup. The specific changes made in network and snort configurations in addition to the `IDS` scenario (refer to Sec III-A) are as follows:

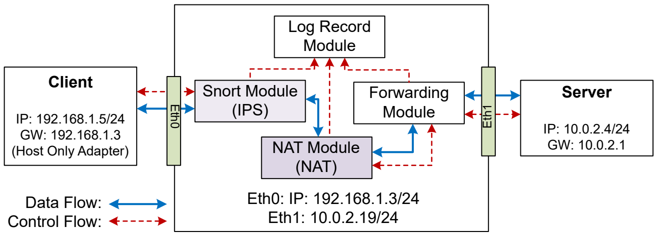


Fig. 4: Scenario 2: IPS between client and server

```
# to inspect each packet with NFQUEUE
iptables -I FORWARD -j NFQUEUE\
--queue-num=4

# Changes in snort.conf
config daq: nfq
config daq_mode: inline
config daq_var: queue=4
```

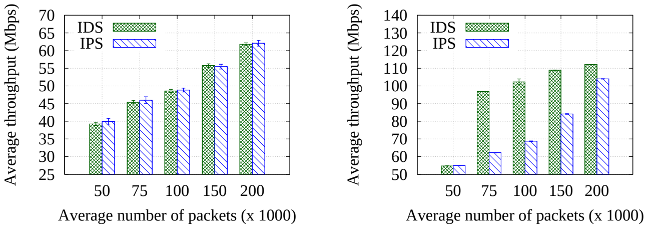
IV. Results and Discussion

We generate TCP and UDP applications using the D-ITG traffic generator [27]. The traffic generator follows the Poisson distribution to generate traffic, in which we vary the average number of packets. Furthermore, each experiment runs 20 times for 30 seconds each. We use the 95% confidence interval [31] to plot the results with a varying average number of packets with payload size 512 bytes. Furthermore, as discussed in Sections III-A and III-B, two scenarios are considered, called `IDS` and `IPS`, respectively. The following performance metrics are considered – average network throughput and jitter for TCP and UDP applications. Whereas latency and the percentage of packet drops are considered for UDP applications as these are the critical parameters for real-time applications. Results obtained using virtual machine-based and containerized deployment are discussed alongside to provide a comparative study on the efficacy of the environments and impact of softwarized security functions on the network performance.

A. Impact on Throughput

1) TCP Application: Figure 5 presents the average throughput for TCP applications tested in different network scenarios. We observe that the throughput increases for both `IPS` and `IDS` with an increase in the average number of packets. Average throughput for TCP applications is higher in case of containerized environment as compared to VMs. Containers share the host operating system kernel, reducing the overhead compared to VMs. The throughput in `IDS` remains relatively high as the packet processing is not affected by the IDS module. However, `IPS` experiences degraded throughput performance. This is because the active threat prevention mechanisms require stateful packet inspection, which is more resource-intensive.

2) UDP Application: Figure 6 presents the average throughput for UDP applications tested in different scenarios for VMs and containers. We see that the average throughput is degraded in case of `IPS` when compared with `IDS`. Furthermore, containerized implementation provides improved

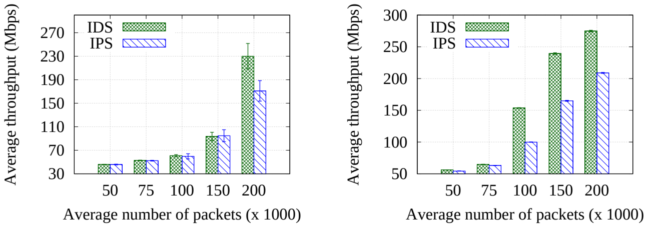


(a) VMs

(b) Containers

Fig. 5: TCP application: Throughput with varying number of packets

throughput as compared to VMs due to the combination of factors related to resource efficiency and potentially efficient kernel-level packet processing, similar to the results with TCP applications. The average throughput for TCP applications is lower than the UDP applications due to less overhead in UDP.



(a) VMs

(b) Containers

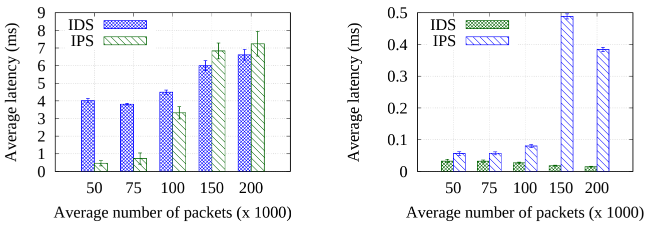
Fig. 6: UDP application: Throughput with varying number of packets

B. Impact on Latency

We measure the impact on latency with varying average number of packets in the network. Figure 7 shows the average latency experienced by packets for both VM-based and containerized implementations with IDS and IPS for UDP applications.

Figure 7 reveals a pronounced increase in latency with large number of packets in the network. This increased latency is a direct consequence of NAT overloading and packet inspection module. The impact on latency in IDS scenario is less as compared to IPS. This is because of additional packet inspection process in IPS that adds additional delay, which, in turn, produces increased latency in case of IPS as compared to IDS. The IDS's passive monitoring allows quicker packet delivery, while IPS introduces delays due to its active intervention even without connection establishment in UDP. However, IPS incurs a lower latency with that of the IDS in the presence of small number of packets. This is because the separate NFQ is assigned for packet processing that does not get overloaded with small number of packets.

Latency observed in case of containers is negligible for smaller number of packets as compared to higher number of packets. However, overall latency for UDP application in case of containers is very less as compared to VMs. This is



(a) VMs

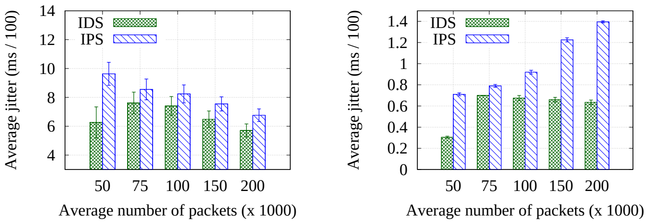
(b) Containers

Fig. 7: UDP application: Latency with varying number of packets

attributable to combination of better packet processing and efficient resource utilization in containerized environment (as compared to VMs) in addition to the connection-less feature of UDP.

C. Impact on Jitter

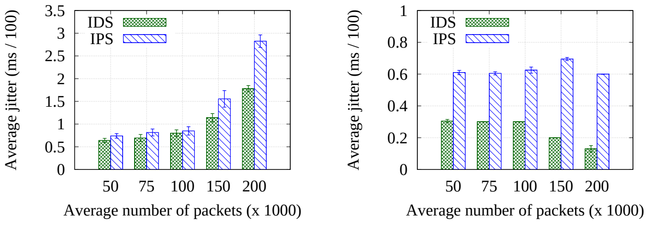
1) TCP Application: We measure the jitter with TCP applications, as it is important for many live applications with stringent reliability requirements, such as remote healthcare. Figure 8 shows the average jitter experienced by packets for TCP applications. IPS processing variability leads to higher jitter due to the dynamic analysis of each packet. Impact of jitter was more in case of VMs than in containers similar to the throughput and latency.



(a) VMs

(b) Containers

Fig. 8: TCP application: Jitter with varying number of packets



(a) VMs

(b) Containers

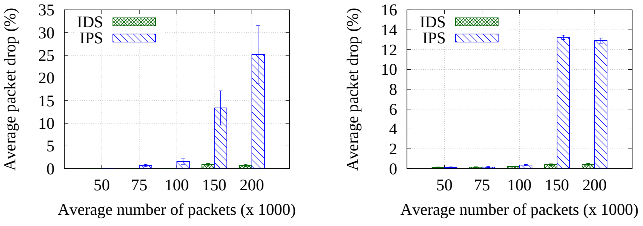
Fig. 9: UDP application: Jitter with varying number of packets

2) UDP Application: Jitter is also important for many fault-tolerant live applications, such as audio-video conferencing. An increased jitter is observed under IPS conditions compared to IDS for UDP applications, as shown in Figure 9. This can be attributed to the variable and intensive packet analysis required by IPS functions, introducing fluctuations in handling times. In contrast, IDS processing remains relatively uniform, leading to a lower jitter. While UDP's lack of acknowledgment mechanisms mitigates the impact on jitter under IDS but does not fully shield it from the interruption and inspection processes of IPS, hence, increased jitter is observed. This emphasizes the non-negligible impact of software-based security functions on jitter, particularly for real-time applications. Similar pattern is observed for containers with respect to jitter in IDS and IPS scenarios. Furthermore, VMs experience higher jitter compared to containers. This is due to the additional layer of abstraction and resource constraints associated with virtualization.

D. Impact on Packet Drop

We present the percentage of packet drops in UDP applications with different security scenarios for VMs and containers in Figure 10. Similar to the throughput, latency, and jitter, we see a degraded network performance in terms of packet loss in the presence of softwarized security functions.

The percentage of packet-drop is higher in case of IPS than that of the IDS. Furthermore, the percentage of packet drop increases with increasing number of packets in the network similar to the latency. This is due to the overloading of the IPS and NAT network functions with large number of packets.



(a) VMs

(b) Containers

Fig. 10: Percentage of packet drops for UDP applications

Figure 10(b) presents the percentage of packet drops in the containerized environment. We see the similar performance trends as in VMs with a reduced percentage of packet drop. Considering the reliability requirements presented in Table I and packet drop with UDP applications in Figures 10(a) and 10(b), TCP should be used for achieving high reliability of the underlying use-case scenarios. However, UDP can be used for increased throughput with low-reliability requirements.

In summary, we see that the softwarized security functions are capable of achieving the diverse QoS requirements in terms of throughput and latency for applications with moderate latency requirements. However, it may not be suitable for ultra-low latency requirements. Furthermore, redundant placement of security functions is also required to achieve high reliability. We note that the network performances may vary depending on the commodity hardware configuration. Therefore, the performance of the softwarized security functions and QoS requirements of associated applications must be taken into consideration while placing them on commodity hardware. We also understand that performance of security functions in terms of impact on QoS parameters is found to be less in containerized environment as compared to VM-based implementation.

V. Use-Case Scenario: 5G Core Network

To study the network performance in a real-world use-case scenario, we consider the standalone architecture of 5G network, as presented in Figure 11. We deploy the 5G core network using Open5GS (https://open5gs.org/open5gs/) open-source software platform. The 5G RAN is deployed using UERANSIM (https://github.com/aligungr/UERANSIM) open-source software platform. UERANSIM and Open5GS are integrated together to have an end-to-end softwarized 5G network. We setup the experiment platform in a host machine with the similar configuration as mentioned in Section III. We create three guest machines, one for UE and gNB placement, another for 5G core network placement with security functions, and the other acts as a server to the UE, similar to the setup explained in Section III. The components inside IDS and IPS modules are similar as discussed in Section III.

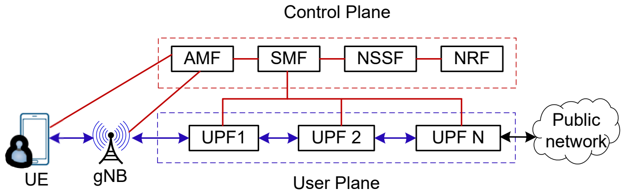
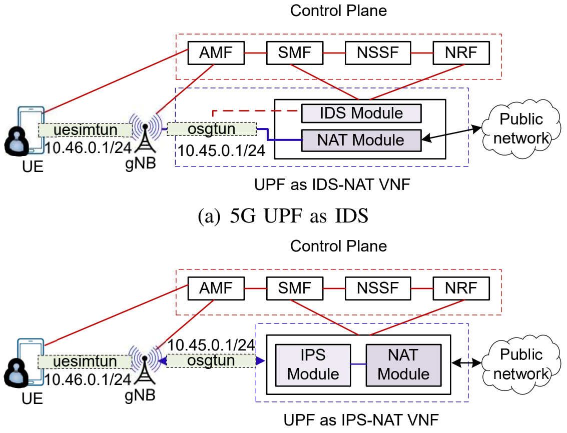


Fig. 11: Standalone architecture of 5G network



(a) 5G UPF as IDS

(b) 5G UPF as IPS

Fig. 12: 5G UPF as IDS and IPS

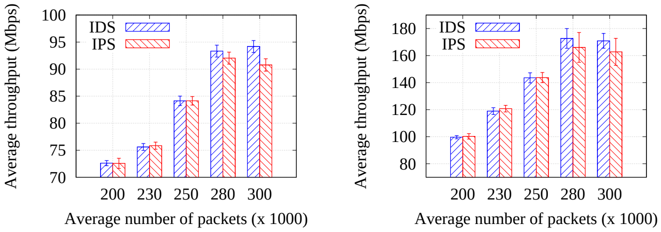
We configure the UPF so that it acts as either IDS or IPS along with NAT. Similar to the generalized setup of the network, NAT functionality is also an integral part of the 5G network system. We consider the same configurations scenarios for IDS and IPS, as described in Section III.

A. Results and Discussion

We present the results on network performance in terms of throughput, latency, and packet drop similar to Section IV.

Figure 13 presents the average network throughput with varying number of packets for TCP and UDP applications.

We see that the average throughput increases linearly initially with less number of packets for both TCP and UDP applications. However, it gets saturated for large number of packets due to the network capacity constraint. As expected, the UDP provides higher throughput than the TCP, as shown in Figures 13(a) and 13(b). Furthermore, we see that IDS and IPS yield a similar throughput for small number of packets. However, IDS yields a higher throughput than IPS with large number of packets. This is because the packets are forwarded without sending them to the NFQ module in IDS. The Snort module only listens to the 5G tunnel interface and sends alerts based on the rule-set. In contrast, the packets are always sent to the NFQ module and analyzed before being forwarded/dropped, when the security function works as IPS. Therefore, the queue is overloaded in the presence of large number of packets. Hence, packets get dropped, which, in turn, leads to decreased network throughput.

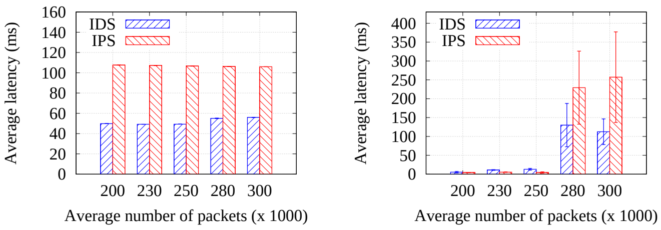


(a) TCP application

(b) UDP application

Fig. 13: Throughput with varying number of packets

1) Latency: Figures 14 presents the network latency with varying number of packets. Similar to the throughput, the latency experienced by TCP applications (refer to Figure 14(a)) is more than the UDP applications (refer to Figure 14(b)). This is because TCP is connection-oriented, which incurs additional delay compared to UDP. Furthermore, the latency experienced by TCP applications is almost constant irrespective of the number of packets in the network, as presented in Figure 14(a). This is due to the fact that TCP adaptively adjusts the congestion-window size, which eventually affects the throughput, as presented in Figure 13. On the other hand, the latency experienced by UDP applications is very low for small number of packets. However, the latency increases non-linearly with large number of packets due to congestion at the NFQ module. This also leads to packet drop, which is discussed in the next section.



(a) TCP application

(b) UDP application

Fig. 14: Latency with varying number of packets

2) Packet Drop: Figure 15 presents the percentage of packet drop for UDP applications with varying number of packets in the network. We see that the percentage of packet drop is very low in the presence of small number of packets for both IDS and IPS. However, it increases non-linearly with large number of packets. This is because of the NAT and NFQ overloading, which causes increased packet drop. Furthermore, the percentage of packet drop is higher in IPS than IDS due to the NFQ overflow.

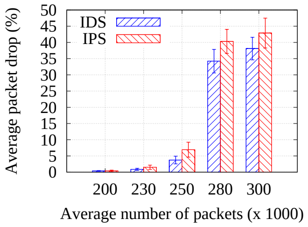


Fig. 15: Percentage of packet drop with varying number of packets for UDP application

In summary, we observe a similar network performance in 5G network enabled with the softwarized security functions.

VI. Conclusion and Implications to Future Work

We studied the performance of network security functions when placed on commodity hardware. We provided thorough implementation details of a secure softwarized network with IDS and IPS. The impact on the network performance is studied in the presence of TCP and UDP applications for VM-based and containerized implementation. Extensive experiment results showed that softwarized network functions significantly impact the network performance in terms of throughput, latency, jitter, and packet drop. Furthermore, it is observed that the containerized placement offers improved performance than VM-based implementation. We also applied the implementation in a 5G network as a use-case scenario. We observed the similar pattern in the results on throughput, latency, and packet drop.

We identify some future research implications as more study is required to understand different aspects of softwarized networks.

Platform to place software functions: As evident from this study and the existing works, the hardware and software platforms, on which the softwarized functions are placed, also affect the network performance. Therefore, further study is required with different hardware and software platforms to have *optimized* softwarized network designed for application-specific service provisioning.

Software tools to create network security functions: In this work, we used Snort to create IDS and IPS security functions and analyzed the impact on network performance. Other software platforms, such as Suricata (https://suricata.io/), can also be used to conduct a comparative study. We note that a few works presented a comparative study between the Snort and Suricata to create the IDS. However, they mainly focused on the efficiency of the platform without considering the impact on the network performances.

Use of TCP and UDP applications: We observed a high throughput and high percentage of packet drop for UDP applications when a large number of packets are generated in the network. On the other hand, TCP yields lower throughput, high latency, and reliable packet delivery. Furthermore, the use of TCP and UDP not only depends on the underlying applications, but also on the network setup. Consequently, in-depth study on the trade-offs between packet-size, average number of packets generated by an application, and the protocol in-use (UDP or TCP) is required for QoS-guaranteed service provisioning.

Application-specific network slicing: With the introduction of network slicing [32], we can create multiple logical networks, each serving different applications based on their service-level agreements. Therefore, in addition to the security functions listed in this work, the performance of each slice should be studied in the presence of different user plane functions such as packet routing and forwarding, policy enforcement, video optimizer, and network monitoring. In such a case, studying the end-to-end network performance of each slice is required.

References

[1] M. Shivank and S. Bera, "Security-as-a-Function in 5G Network: Implementation and Performance Evaluation," in *IEEE International Conference on Signal Processing and Communications (SPCOM)*, Bangalore, India, 2024, p. 5.

[2] J. Navarro-Ortiz, P. Romero-Diaz, S. Sendra, P. Ameigeiras, J. J. Ramos-Munoz, and J. M. Lopez-Soler, "A survey on 5G usage scenarios and traffic models," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 2, pp. 905–929, 2020.

[3] "Verticals uRLLC use cases and requirements," NGMN Alliance, Tech. Rep. V 2.5.4, Feb. 2020.

[4] "5G: Service requirements for the 5G system," 3GPP, Tech. Rep. TS 22.261, Version 15.9.0, Release 15, 2021.

[5] W. Xian, K. Yu, F. Han, L. Fang, D. He, and Q.-L. Han, "Advanced manufacturing in Industry 5.0: A survey of key enabling technologies and future trends," *IEEE Trans. Ind. Informat.*, pp. 1–15, 2023.

[6] Y. Jarraya, T. Madi, and M. Debbabi, "A survey and a layered taxonomy of software-defined networking," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 4, pp. 1955–1980, 2014.

[7] B. A. A. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka, and T. Turletti, "A survey of software-defined networking: Past, present, and future of programmable networks," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 3, pp. 1617–1634, 2014.

[8] J. d. J. Gil Herrera and J. F. Botero Vega, "Network functions virtualization: A survey," *IEEE Lat. Am. Trans.*, vol. 14, no. 2, pp. 983–997, 2016.

[9] H. Farhady, H. Lee, and A. Nakao, "Software-defined networking: A survey," *Computer Networks*, vol. 81, pp. 79–95, 2015.

[10] D. Gedia and L. Perigo, "Performance evaluation of SDN-VNF in virtual machine and container," in *IEEE NFV-SDN*, Nov. 2018.

[11] W. Cerroni, A. Galis, K. Shiomoto, and M. F. Zhani, "Network softwarization and management," *IEEE Commun. Mag.*, vol. 58, no. 10, pp. 14–15, 2020.

[12] 3GPP, "5G System Overview."

[13] S. Agarwal, F. Malandrino, C.-F. Chiasserini, and S. De, "Joint VNF placement and CPU allocation in 5G," in *IEEE INFOCOM*, Apr. 2018, pp. 1943–1951.

[14] L. Liu, S. Guo, G. Liu, and Y. Yang, "Joint dynamical VNF placement and SFC routing in NFV-enabled SDNs," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 4, pp. 4263–4276, Dec. 2021.

[15] A. Gupta and L. S. Sharma, "Performance analysis and comparison of Snort on various platforms," *Int. J. Comput. Inf. Syst. Ind. Manag. Appl.*, vol. 10, pp. 023–032, 2020.

[16] T. Salah, M. J. Zemerly, C. Y. Yeun, M. Al-Qutayri, and Y. Al-Hammadi, "Performance comparison between container-based and VM-based services," in *IEEE Conf. on Innovations in Clouds, Internet and Networks*, Mar. 2017.

[17] Z. Wang, "Can "Micro VM" become the next generation computing platform?: Performance comparison between light weight Virtual Machine, container, and traditional Virtual Machine," in *IEEE Intl. Conf. on Computer Science, Artificial Intelligence and Electronic Engineering*, Aug. 2021.

[18] S. Gallenmüller, J. Naab, I. Adam, and G. Carle, "5G QoS: Impact of security functions on latency," in *IEEE/IFIP NOMS*, Apr. 2020, pp. 1–9.

[19] J. E. Varghese and B. Muniyal, "An efficient IDS framework for DDoS attacks in SDN environment," *IEEE Access*, vol. 9, pp. 69 680–69 699, 2021.

[20] R. Shams, D. O. Suri, F. Hanif, and P. Otero, "Comparative analysis of intrusion detection systems in SDN," in *Global Conference on Wireless and Optical Technologies*, 2023, pp. 1–9.

[21] J. Steinberger, A. Sperotto, M. Golling, and H. Baier, "How to exchange security events? Overview and evaluation of formats and protocols," in *IEEE/IFIP IM*, 2015, pp. 261–269.

[22] D. Fadhilah and M. I. Marzuki, "Performance analysis of IDS Snort and IDS Suricata with many-core processor in virtual machines against Dos/DDoS attacks," in *Intl. Conf. on Broadband Communications, Wireless Sensors and Powering*, Sep. 2020, pp. 157–162.

[23] S. Seeber, L. Stiemert, and G. D. Rodosek, "Towards an SDN-enabled IDS environment," in *IEEE Conf. on Communications and Network Security*, 2015, pp. 751–752.

[24] V. Aggarwal and B. Thangaraju, "Performance analysis of virtualisation technologies in NFV and edge deployments," in *IEEE CONNECT*, Jul. 2020.

[25] Z. Li, M. Kihl, Q. Lu, and J. A. Andersson, "Performance overhead comparison between hypervisor and container based virtualization," in *IEEE AINA*, Mar. 2017.

[26] S. K. Garg, J. Lakshmi, and J. Johny, "Migrating VM workloads to containers: Issues and challenges," in *IEEE CLOUD*, Jul. 2018.

[27] A. Botta, A. Dainotti, and A. Pescapé, "A tool for the generation of realistic network workload for emerging networking scenarios," *Computer Networks*, vol. 56, no. 15, pp. 3531–3547, 2012.

[28] W. Kellerer, A. Basta, P. Babarczi, A. Blenk, M. He, M. Klugel, and A. M. Alba, "How to measure network flexibility? A proposal for evaluating softwarized networks," *IEEE Commun. Mag.*, vol. 56, no. 10, pp. 186–192, 2018.

[29] W. Felter, A. Ferreira, R. Rajamony, and J. Rubio, "An updated performance comparison of virtual machines and Linux containers," in *IEEE ISPASS*, Mar. 2015.

[30] "NFV Release 2 Description, v1.13.0," Dec. 2021.

[31] A. Hackshaw, *Statistical Formulae for Calculating Some 95% Confidence Intervals. A Concise Guide to Clinical Trials*. John Wiley & Sons, Ltd, 2009.

[32] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini, and H. Flinck, "Network slicing and softwarization: A survey on principles, enabling technologies, and solutions," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 2429–2453, 2018.