Processing Time Comparison of a Hardware-Based Firewall and its Virtualized Counterpart

Steffen Gebert, Alexander Müssig, Stanislav Lange, Thomas Zinner, Nicholas Gray, Phuoc Tran-Gia

University of Würzburg, Institute of Computer Science,
Am Hubland, 97074 Würzburg, Germany
{steffen.gebert,alexander.muessig,stanislav.lange,zinner,nicholas.gray,trangia}@informatik.uni-wuerzburg.de

Abstract. The network functions virtualization (NFV) paradigm promises higher flexibility, vendor-independence, and higher cost-efficiency for network operators. Its key concept consists of virtualizing the functions of specialized hardware-based middleboxes like load balancers or firewalls and running them on commercial off-the-shelf (COTS) hardware.
This work aims at investigating the performance implications that result from migrating from a middlebox-based hardware deployment to a NFV-based software solution. Such analyses pave the way towards deriving guidelines that help determining in which network environments NFV poses a viable alternative to today's middlebox-heavy architectures. To this end, a firewall is chosen as an exemplary network function and a performance comparison between a dedicated hardware device and a commercially distributed virtualized solution by the same vendor is drawn. This comparison focuses on the packet delay, while varying the load level that is applied to the network function under test. Based on traffic measurements of a university campus network, conclusions regarding possible fields of application are drawn.

1 Introduction

Today's networks rely heavily on specialized, hardware-based middleboxes for a multitude of networking tasks such as firewalling or load balancing. Despite their advantages with respect to performance, these specialized middleboxes exhibit drawbacks in terms of acquisition costs, flexibility, and vendor-dependence, favoring a paradigm shift towards NFV. By leveraging virtualization techniques, the NFV concept allows migrating the functionality of the specialized middleboxes to software running on COTS hardware. On the one hand, this reduces the high acquisition costs associated with the former. On the other hand, network operators can benefit from vendor independence and an increase in flexibility.

Before deploying virtualized network functions (VNFs) in an existing network, its operator needs to determine whether the resulting system can still cope with the load offered by the network. Due to the fact that different network

functions behave differently and have specific requirements, such a question can not be addressed in a general way. Hence, this work focuses on the performance evaluation of a firewall which is commercially available both as a hardware entity as well as a VNF. In particular, Cisco's *Adaptive Security Appliance Service Module* (ASASM)[1] and its virtualized counterpart *Adaptive Security Virtual Appliance* (ASAv)[2] are utilized in our experiments. Based on traffic statistics of a university campus network, the feasibility of the two deployment types in this environment is investigated.

Measurements focus on the packet processing time while varying the system load in terms of number of concurrent sessions. By using a dedicated hardware-based traffic generator which applies realistic load profiles to the device under test (DuT), such an analysis sheds light on the feasibility of NFV-based solutions in realistic environments and thus, can contribute towards identifying scenarios in which adopting the NFV paradigm makes the most sense.

This work is structured as follows. Section 2 provides an overview of related work regarding the general topic of VNF benchmarking as well as performance evaluation of virtualized firewalls in particular. In Section 3, the testbed setup is presented alongside the different scenarios and parameters that are used in this work. After discussing the results of the measurements in Section 4, Section 5 concludes the paper.

2 Related Work

While different network functions have different characteristics in terms of their behavior and requirements, there is a common ground when it comes to evaluating their performance. RFC 2544 [3] provides benchmarking guidelines for networking devices such as routers, switches, or firewalls. These guidelines include key performance indicators for various DuTs as well as the methodology for measuring latency and throughput of these devices. Several additional documents were released in order to take into account the increased set of features and capabilities of network elements [1, 2].

When attempting to evaluate the performance of virtualized network functions that run on COTS hardware, additional challenges need to be addressed. Due to the additional abstraction layer introduced by the softwarization of network functions, system performance does not only depend on the underlying hardware but also on the particular VNF implementation [5]. In contrast to ASIC-based packet processing, effects like scheduling and caching also impact the predictability and reliability of software solutions. Furthermore, interdependencies between VNF instances running on the same physical substrate can affect the system's behavior [6, 7].

[1] `http://www.cisco.com/c/en/us/products/interfaces-modules/catalyst-6500-series-7600-series-asa-services-module/index.html`

[2] `http://www.cisco.com/c/en/us/products/collateral/security/adaptive-security-virtual-appliance-asav/datasheet-c78-733399.html`

For the specific case of firewall benchmarking, a methodology is presented in RFC 3511 [4] that characterizes the performance of a firewall and describes formats for presenting benchmarking results. The measurements conducted in the remainder of this paper follow this RFC as a guideline.

A performance comparison between a Cisco ASA 5505 hardware and the software-based Linux *iptables* is presented in [9]. The work focuses on three main performance metrics: throughput, latency, and concurrent connections. While the Cisco ASA outperforms Linux iptables in terms of throughput and latency, the latter is capable of handling bursts of packets better and achieves a higher number of concurrent sessions.

3 Methodology

In order to investigate whether a virtualized firewall can replace its more expensive and inflexible hardware-based counterpart in a particular scenario, we evaluate and compare their performance in a dedicated testbed. First, the testbed setup is presented alongside the specifications of its hardware and software components. Second, the course of experiments for the aforementioned performance evaluation as well as parameters and performance indicators are described.

3.1 Measurement Setup

Figure 1 shows the main components of the testbed utilized in this work. It is comprised of two networks, an external and an internal network, which are separated by the firewall. The networks are represented by two switches that are connected to the firewall and different types of traffic are generated using two Spirent C1[3][4] traffic generators. The Spirent C1 is a dedicated FPGA-based traffic generator equipped with four 1 GbE interfaces and allows highly accurate and reliable measurements of various performance indicators like packet delay.

The first traffic generator produces stateful TCP traffic using the Avalanche software. In this context, "stateful" refers to the fact that the firewall needs to keep track of the state of each connection according to the TCP state machine. This traffic is used to set the DuT under different load levels in terms of varying numbers of active connections. The second C1 device is controlled by the TestCenter software in order to benefit from the highly precise hardware-based delay measurement that, however, only supports stateless traffic. Probe packets are sent once per second through a TCP connection that was previously opened by emulating the TCP handshake.

As mentioned earlier, two different firewall deployments are analyzed in this work and detailed in Table 1. The first component is the Cisco *Adaptive Security Appliance Service Module* (ASASM), a hardware solution. While layer 3 and 4

[3] `http://www.spirent.com/ethernet_testing/platforms/c1_chassis`

[4] `http://www.spirent.com/~/media/Datasheets/Broadband/PAB/SpirentTestCenter/STC_C1-Appliance_Datasheet.pdf`

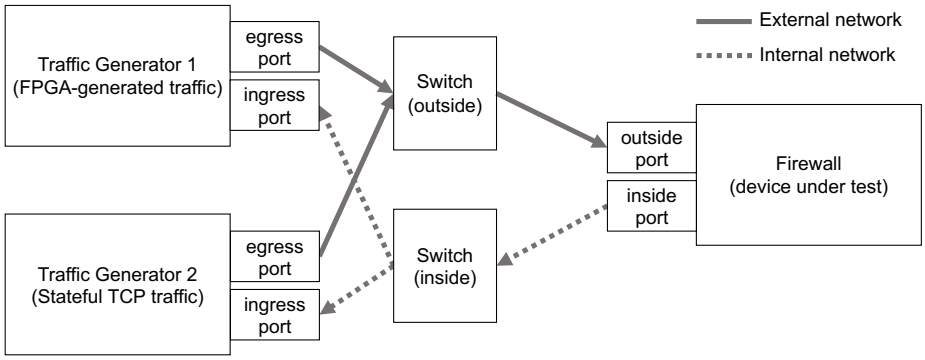

Fig. 1. Hardware setup used in this work.

processing is performed in ASIC / TCAM, the general purpose Xeon CPUs of the ASASM are used for tasks like VPN, content inspections, and management. The Cisco *Adaptive Security Virtual Appliance* (ASAv) ASAv30 runs virtualized using VMware ESXi on a Cisco *UCS* server and is configured as per vendor recommendations. While the data sheet of the ASASM provides basic information on the processing delay of the appliance, no information is available in case of the ASAv. This work sheds some light on the performance differences between these commercially available solutions.

In order to achieve realistic testing conditions, around 1300 rules are configured on the firewalls. These rules correspond to those installed on our campus firewall. Hence, the resulting measurements can provide insights regarding the feasibility of the two firewall deployment types in such a network.

3.2 Experiments

Using the testbed setup described in the previous section, the performance of the two firewall types is evaluated with respect to the achieved processing time. This section presents the methodology for investigations regarding the effects of different load levels on the processing time of TCP packets. The test scenarios were developed together with firewall administrators and campus network administrators at the computing center of our university.

The main test case investigates the processing times of TCP packets while the firewall is exposed to various amounts of load in terms of open TCP connections. Before the measurement is started, the TCP connections are opened by utilizing Avalanche in order to simulate users in the internal network that request file downloads from servers in the external network (cf. Figure 1). After downloading a small 1 kilobyte file, the TCP connection is left open and users keep on requesting other files from different servers until the desired number of TCP connections is established. In order to avoid completely idle connections, a 1 kilobyte file is requested via HTTP for each connection every 10 seconds. The total number of active connections is varied between 1,000 and more than

Product		
Vendor	Cisco	Cisco
Series	ASASM	ASAv
Model	WS-SVC-ASA-SM1	ASAv30 v9.4(1)3
Deployment Type	Physical, Switch Blade	Virtual
CPU	2x Xeon 5600 2.00 GHz, 6 cores	4 vCPUs
Memory	24 GB	8 GB
Pricing		
Perpetual[5]	$97,750	$15,980
On-demand[6]	-	$1.39 per hour
Hardware Base Platform		
Platform	Catalyst 6509 Switch	Cisco UCS C220 M3
CPU	VS-S720-10G, 600 MHz	Xeon E5-2680, 8 cores
Memory	-	64 GB DDR3
Hypervisor	-	VMware ESXi 5.5
Specifications		
Max. throughput	20 Gbps	2 Gbps
Connections/sec	300,000	60,000
Concurrent conn.	10,000,000	500,000

Table 1. Devices under test.

500,000 in order to cover a diverse set of scenarios. Finally, the TestCenter software is used to instruct the FPGA to generate one TCP packet per second and capture the resulting packet delays.

4 Results

As the firewall is the access gate to a network where all traffic from or to this network passes, it can become a bottleneck between the internal and external network. Especially the software implementation of the firewall raises the question whether it can keep pace with its hardware counterpart. Therefore, the performance of the two firewall deployment types according to the methodology described in Section 3 is evaluated in this chapter. Each experiment lasts 5 minutes and is repeated 5 times.

4.1 Processing Time

In Figure 2, an overview of the distribution of processing times of the ASASM is provided. While the x-axis displays the processing times, the y-axis represents the fraction of observations in which the corresponding value was not exceeded.

[5] according to `http://ciscoprice.com/`
[6] on Amazon AWS

Furthermore, differently colored curves mark different load levels in terms of concurrent connections. For the sake of readability, only the most relevant range of x-values is depicted.

There are three main observations. First, the processing time has a low variability for all scenarios, i.e., an interquartile range of roughly 4 microseconds with values between 60 and 85 microseconds. Second, processing times show a strictly decreasing behavior with increasing numbers of concurrent connections. This phenomenon could stem from interrupt mitigation mechanisms similar to those of the *New API* [8], the network I/O API that is utilized by Linux operating systems. In order to decrease the overhead that results from each individual packet causing an interrupt, this mechanism accumulates packets until either a certain amount of packets is collected or processing is initiated by a timeout. Hence, lower load levels cause higher delays as packets need to wait for the timeout to trigger processing, while packets exceed the threshold at an increasing rate in case of high load levels, resulting in lower delays. Finally, the small gap between the two highest load levels indicates a converging behavior with respect to processing times.

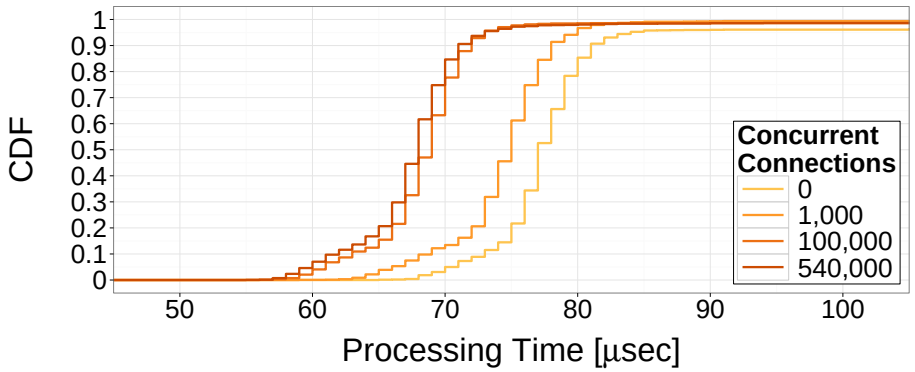

Fig. 2. Processing times achieved by the ASASM.

Similar to the previous figure, Figure 3 displays the distribution of processing times achieved by the virtualized firewall ASAv. Again, the x-axis is limited to the most relevant interval. In contrast to the maximum number of connections of 540,000 that was used in the context of the ASASM-related measurements, the maximum for the ASAv-related measurements is at 500,000. The reason for this parameter choice is a hard-coded capacity limit implemented in the ASAv software (cf. Table 1).

The ASAv exhibits two modes regarding the processing times. For low numbers of TCP sessions, higher delays and a significantly higher variance are observed. In these cases, the ASAv introduces packet delays between 100 and 800 microseconds, roughly ten times higher than when using an ASASM. When increasing the load, however, the variance and the absolute values decrease and

stabilize in an interval between 100 and 350 microseconds. As in case of the hardware-based ASASM, there seems to be a mechanism that positively affects packet processing delays of the firewall at higher load levels.

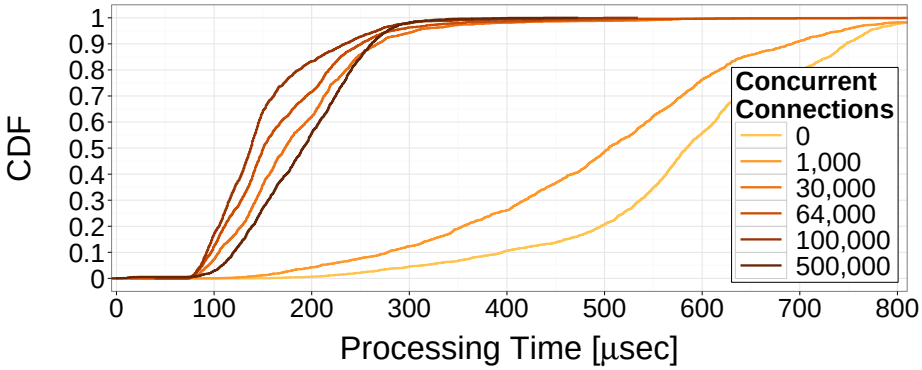

Fig. 3. Processing times achieved by the ASAv.

4.2 Suitability for a Campus Network

In order to put the measurement results into the context of a real network, traffic measurements have been conducted in the university's campus network using NetFlow. During two working days of observation, the highest number of parallel connections at the Internet uplink with a capacity of 2×3.5 Gbps was found to be around 420,000 connections. Further, the highest number of new connections per second was found to be around 6,500-6,800, with two exceptions, during which 14,700 and 20,300 new connections were opened within one second, respectively. At the 10 G connection of a single building (computer science faculty), a maximum of around 177,000 concurrent connections with peaks of up to 4,400 new connections per second were observed.

Given the ability of ASAv to handle up to 500,000 concurrent connections, it seems to be a feasible approach for both scenarios. However, the total throughput of only up to 2 Gbps is the limiting factor, which would restrict the use of ASAv in the investigated network, e.g., to the protection of clusters of servers or buildings connecting only few users through smaller links. An investigation of the maximum number of connections per second that the ASAv can handle remains for future work.

5 Conclusion

This work presents a comparison of processing delays introduced by two types of firewall deployments, namely a hardware-based approach and an NFV-based

virtualized approach. In particular, Cisco's Adaptive Security Appliance Service Module (ASASM) and its virtualized counterpart, the Adaptive Security Virtual Appliance (ASAv), are tested. Both of these are commercially available products. By comparing these two firewall deployment types, this work provides guidance for dimensioning of NFV systems.

All measurements are performed in a testbed featuring two Spirent C1 traffic generators, an industrial-grade packet generator and test platform. The evaluations presented in this paper allow characterizing scenarios in which the more flexible and cost-efficient virtualized approach can provide a sufficient level of performance and thus poses a viable alternative to purchasing specialized hardware.

In all investigated scenarios, using the virtualized ASAv results in an increase of packet delays by a factor of up to ten. Furthermore, a higher variance of the processing times is observed. However, even in the virtualized case, the majority of measured processing times is below 1 millisecond. Depending on the usage scenario, delays in this order of magnitude might be acceptable, e.g., for perimeter firewalls connecting networks to the Internet.

Future investigations with a better-equipped 10 GbE traffic generator could extend this work by providing additional insights into the performance limits of the DuTs. More directions for future work include a comparison between different hypervisors like the KVM-based ASAv, as well as different virtualization strategies such as paravirtualization. Additionally, investigations regarding the load in terms of the number of new connections per second would help identifying performance bottlenecks. Consequently, such analyses can provide guidelines for network operators to decide between a hardware-based solution and a virtualized one.

To further benefit from running virtual instances, e.g., by dynamically scaling up and down, the current version of the ASAv lacks the feature of operating in a cluster. Once available or when investigating alternative implementations, this important aspect of a VNF deployment offers even more room for further measurements.

Acknowledgment

This work has been performed in the framework of the SARDINE project and is partly funded by the BMBF (Project ID 16KIS0261). The authors alone are responsible for the content of the paper.

References

1. R. Asati, C. Pignataro, F. Calabria, and C. Olvera. RFC26201: Device Reset Characterization. IETF, 2011.
2. S. Bradner, K. Dubray, J. McQuaid, and A. Morton. RFC6815: Applicability Statement for RFC 2544: Use on Production Networks Considered Harmful. IETF, 2012.

3. S. Bradner and J. McQuaid. RFC2544: Benchmarking Methodology for Network Interconnect Devices. IETF, 1999.
4. B. Hickman, D. Newman, S. Tadjudin, and T. Martin. RFC3511: Benchmarking Methodology for Firewall Performance. IETF, 2003.
5. S. Lange, A. Nguyen-Ngoc, S. Gebert, et al. Performance Benchmarking of a Software-Based LTE SGW. In *2nd International Workshop on Management of SDN and NFV Systems*, 2015.
6. A. Morton. Considerations for Benchmarking Virtual Network Functions and Their Infrastructure. Internet-Draft draft-morton-bmwg-virtual-net-03, 2015.
7. Overture, Brocade, Intel, Spirent, and Integra. NFV Performance Benchmarking for vCPE. Executive Summary, 2015.
8. J. H. Salim, R. Olsson, and A. Kuznetsov. Beyond softnet. In *Proceedings of the 5th annual Linux Showcase & Conference*, 2001.
9. J. Xu and W. Su. Performance Evaluations of Cisco ASA and Linux IPTables Firewall Solutions. Master's thesis, Halmstad University, 2013.