Annals of Emerging Technologies in Computing (AETiC)
Vol. 4, No. 1, 2020

Research Article

Cloud based Distributed Denial of Service Alleviation System

Michal Zak and J. Andrew Ware*

Faculty of Computing, Engineering and Science, University of South Wales, United Kingdom
michal.zak@southwales.ac.uk; andrew.ware@southwales.ac.uk
*Correspondence: andrew.ware@southwales.ac.uk

Received: 28th November 2019; Accepted: 12th December 2019; Published: 1st January 2020

Abstract: Cloud computing is a phenomenon that is changing information technology, with many companies no longer having data and resources retained within their own premises. Instead they are utilising cloud computing and its centralised resources. There are many benefits of this approach such as pay-per-use model, elasticity of operation and on demand resourcing. However, this approach also introduces additional security challenges. Security involves a triad of considerations, those being confidentiality, integrity and availability, often abbreviated to CIA. This work focusses on the last aspect of the CIA triad – availability, which is even more crucial for cloud-based platforms as centralised resources need to be provided at a distance to the end customers. Several factors including 'denial of service' attack impact availability. Moreover, current protection frameworks do not sufficiently consider the issues of verification, scalability and end-to-end latency. Hence, a new framework has been designed to fill the identified gap. The framework referred to as the cloud-based Distributed Denial of Service Alleviation System (DDoSAS) is based on its predecessor Enhanced DDoS-MS. The new framework has been implemented using Amazon Web Services. The work serves to provide a baseline for measuring end-to-end latency in real-life scenarios.

Keywords: ***Denial of service attack; DoS; DDoS; Cloud computing; availability challenges***

---

1. Introduction

The phenomenon of cloud computing is shifting the information technology landscape. The significance and magnitude of this industry can be illustrated by the prediction made by Gartner that this sector would be worth $411 billion by 2020 [1]. There are multiple reasons why this technology is becoming increasingly popular, including its pay-per-use model, and its scalability and elasticity. However, there are security challenges associated with this new technology, that can be split into the well know CIA triad of confidentiality, integrity and availability. While there are multiple attack vectors that directly or indirectly affect the first and second area, the work reported here focusses on the third area – availability. Availability of the service is crucial to cloud computing, as it is entirely reliant on delivering its services at a distance. Several attack vectors including, but not limited to, denial of service and distributed denial of service attacks, can curtail availability. The work reported here is targeting this subset of availability threats.

Michal Zak and J. Andrew Ware, "Cloud based Distributed Denial of Service Alleviation System", *Annals of Emerging Technologies in Computing (AETiC)*, Print ISSN: 2516-0281, Online ISSN: 2516-029X, pp. 44-53, Vol. 4, No. 1, 1st January 2020, Published by International Association of Educators and Researchers (IAER), DOI: 10.33166/AETiC.2020.01.005, Available: http://aetic.theiaer.org/archive/v4/v4n1/p5.html.

The attacks have a relatively unsophisticated modus operandi, where an adversary seeks to overwhelm the targeted resources with illegitimate requests. Possible resource targets can be the connecting network line and its bandwidth, or a server's CPU resources. At the point when the resource has no capacity left to keep the service operational, the availability is affected, and the service becomes unavailable for use by legitimate users. Despite several proposed frameworks that attempt to tackle this kind of malicious behaviour, no comprehensive research has been conducted that facilitates solutions that deal with the three significant issues of traffic verification, scalability and latency. The solution described here builds on the Enhanced Distributed Denial of Services Mitigation System Framework [2] to fill the gap. This new framework has been named the Cloud based Distributed Denial of Service Alleviation System (DDoSAS).

2. Cloud Computing and Security Context

Although cloud computing involves complex systems, the fundamental idea can be described with a certain level of abstraction as a combination of centralisation and virtualisation approaches. Using a combination of these two approaches the central resource can be optimally utilised to service several customers. The National Institute of Standards and Technology [3] has articulated several characteristics of the cloud: on-demand self-service; broad network access; resource pooling; rapid elasticity; and, measured service.

Cloud computing can be provided as a service using varied combinations of three common models: Infrastructure as a Service – IaaS; Platform as a Service – PaaS; and, Software as a Service – SaaS. IaaS involves providing hardware resources to the subscribed customer. PaaS, is similar to IaaS but in addition to the hardware resource, provides a pre-created environment that can be used directly. SaaS provides a direct service to the customer without them needing to consider explicitly resourcing.

Cloud computing can be deployed in three common scenarios: private, public and hybrid. These environments can be subject to attacks that focus on threats to availability. The current mitigation frameworks do not comprehensively consider the triad of scalability, packet verification and latency [2].

The mitigation techniques can be defined in terms of the place of function: source-based, network-based, or victim-based. Source-based solutions are located near to the source of the traffic, that is close to potential attacker, and have the advantage that the traffic is captured even before it reaches the public network, and therefore the public infrastructure is saved the effort of delivering unsolicited malicious traffic to the targeted server. However, due to the nature of the public network, it requires the significant involvement of all parties that have access to it. Network-based solutions are deployed within the public infrastructure, for example with cooperation of Internet Service Providers, resulting in a reduction in the number of entities involved in the collaboration. Victim-based solutions, deployed close to the subject that might be target of attack, minimise the number of collaborating entities.

While the first categorisation is based on place of application, the second categorisation is based on the time of application. The solution can be proactive or reactive. Proactive means that the solution is not waiting for an attack to start but is active even when there is no attack and mitigates even the possibility of an attack beginning. A reactive solution only responds when an attack is detected.

3. Cloud based Distributed Denial of Service Alleviation System

The new solution is proactive, targeting all traffic regardless of the current status. The framework is active whether dealing with a normal and legitimate load or in an under the attack scenario. The benefit of this approach is that there is no reaction time necessary whereas with a re-active system when an attack is detected there is a latency while response function is initiated. The solution is one that is victim-based.

The solution consists of five sectional functional elements (see Figure 1). The first functional element, the firewall, has the key role of arbiter on whether traffic is allowed to progress. The firewall

uses information about the sources, including the IP source address and TTL value, to determine its decision. The firewall also corelates information about the source, including any malicious payload in packets and suspected abuse by a verified human user.

The second functional element, the verification cluster provides information about the sources through utilization of human detection mechanism. The system used to distinguish between humans and automated systems is CAPTCHA [4]. The third functional element provides an additional level of protection through the Crypto-Puzzle cluster and gives further clarification on the source of the traffic. The fourth element is deep packet inspection and is performed through the Intrusion Prevention System (IPS). The payload can contain malicious content in the packet payload; the payload could cause availability issues, as a single low and slow type of attack, which can significantly strain system handling the request. The fifth functional element of the framework is the reverse proxy that serves as an additional layer of anonymization of the location for the protected resources. The reverse proxy receives the request before forwarding it together with its identification attributes, which means that the location of the protected server is kept hidden. The reverse proxy also verifies users and counts the number of requests in a given time per connection to make sure that the human users are utilizing the service in an expected and legitimate workflow.

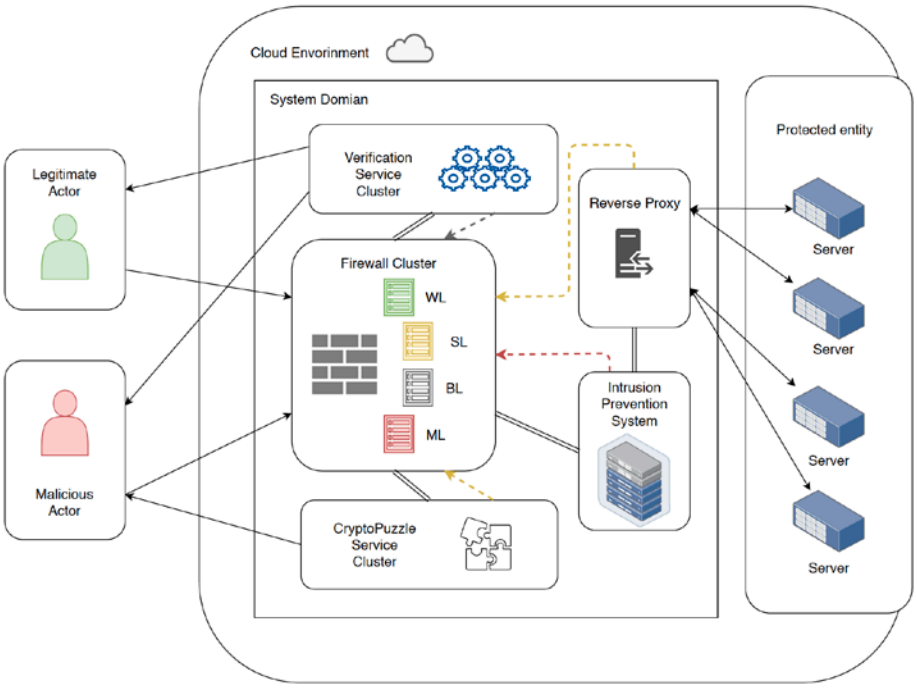

Figure 1. Architecture diagram of Cloud based Districted Denial of Service Alleviation System.

3. 1. Key differences of the Cloud based Distributed Denial of Service Alleviation System

The evolutionary step that led to the improved framework is specified by several modifications that are articulated below.

1. The position for the deployed solution has changed. The preferred location for deployment of the predecessor is the customer's edge network. The intention is to provide the security measures before the traffic even reaches the public connectivity network. When deployed in the customer's edge it can act on network traffic that either originates from that network directly or that is passing through that part of the network. The challenge that it poses is that this security protection framework will not see network communication that does not satisfy either of those two criteria. A customer that is subscribed to the cloud service can work from any location in the world, however if the protection is deployed at customer's edge it would negate this inherent type of cloud computing benefit. Users that are not within the dedicated

network would, in effect, not be covered by the framework's protection and could be the source of the attack against the protected resource. The use-case includes situations when the framework is both effective and not effective within the defined scope. The outcome of the non-effective situation is one that describes a malicious intention of a user that is part of the customer edge network. In this case the advisory would be able to use an option to utilise tunnelling solutions that would encrypt all the traffic and send it to the exit node, which would be outside of the guarded customer's edge network. That would lead to the previous scenario where the malicious actor is effectively any other person outside of the customer's edge network. If the cloud network edge only accepted traffic from that customer's network edge, it would mean that the accessibility and availability of the cloud would be restricted.

Based on the conceptual analysis of the scenarios it is necessary to evolve this solution and change its placement to the cloud provider's edge. This would satisfy both scenarios considered earlier, where the solution would be restricted to only the customer's edge network or can be opened to the allow the wider public accessibility, thereby not restricting the accessibility and availability of the cloud resources to the subscriber's network.

2. The technological stack that is used to build the framework has been changed. Changing the concept of the place of deployment allows the framework to be run on a virtual platform directly in the cloud. Virtualisation significantly reduces the initial costs of framework deployment compared to the on-premise solution. The technological advancement connected to the stack technology is a point of replication. Replication can be done through infrastructure as a code approach. Therefore, not just the framework logic but the framework itself can be implemented via a software solution. Replication can be achieved via software defined automation, which can be accommodated within the cloud computing environment. The new framework is built with the replication and automation factors in mind. Therefore, the framework can work with single tenant as well as multi-tenant architectures. A single tenant solution could be further divided into two options. The first would mean that the framework would be restricted to the network subset of a customer, and the customer network edge network, while the second would be similar but not impose the origin of the customer network end subnet. The multi-tenant architecture would allow a single framework for multiple customers.
3. Redundancy has been improved at two levels: that is, through improvements to the separate elements and to the overall framework. The new framework provides capability for both single-tenant and multi-tenant scenarios. Redundancy improvements at framework level can be achieved through automation and software defined infrastructure, which leads to possible replication and redundancy of the framework for each customer. The improvement to separate framework elements provides the redundancy and resiliency in a sense of the components themselves. A framework element can automatically replace itself in case of possible health failure. The contextual analysis shows that the change is desirable for the front facing components that could pose a single point of failure.

4. Experimental results evaluation

The framework has been implemented using Amazon Web Services (AWS cloud), a rapidly expanding provider [5]. To facilitate cross comparable results the framework was benchmarked against its predecessor. Moreover, to achieve a verification on a higher scale, the scale was intentionally increased in both metrics of quantity of the traffic that the framework had to handle and the intensity of the load that sent into the framework.

4.1. Cross comparison experiments

Three distinct experiments were designed to investigate dynamic scenarios that involved varying the traffic intensity within each traffic load being sent to the framework. The first focused on

increase in the traffic intensity, the second on a situation where there is traffic fluctuation, the third where there is a significant increase and decrease of the traffic intensity. Each cross-comparison scenario consisted of one thousand ICMP requests within each stream. Therefore, each network stream consisted of two thousand network packets.

4.1.1 First cross dynamic comparison experiment

The results for the defined load are shown in Figure 2. The solid line represents the response time and the dashed line the time for processing the ICMP responses. There is a significant increase for ICMP request number 2084 which had a response time 0. 543ms. This is an exceptional anomaly that might be cause by other data passing through the public environment.

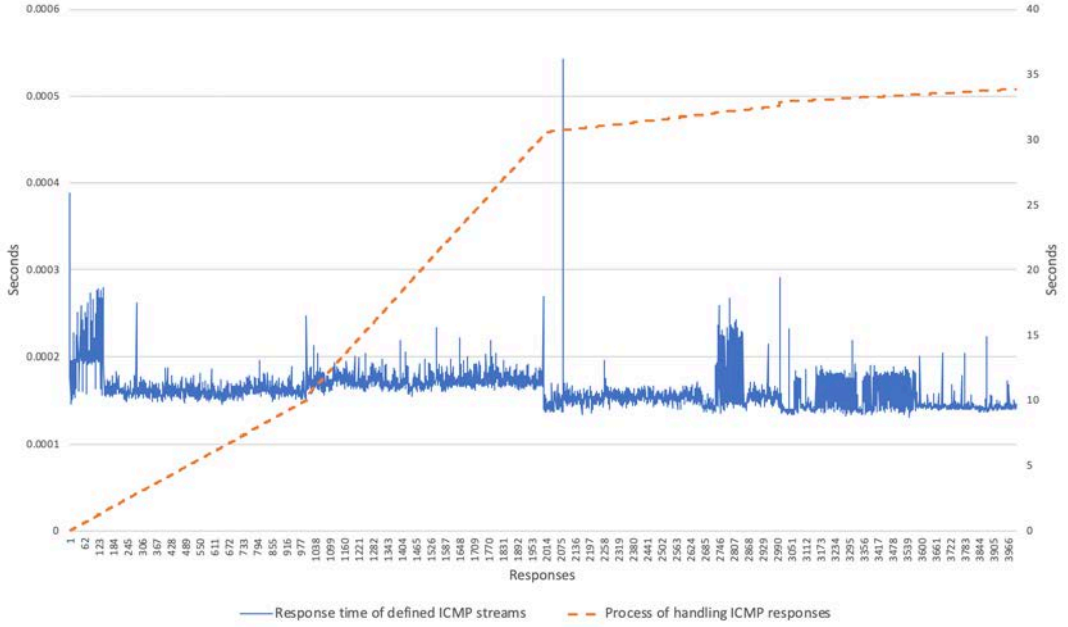

Figure 2. Results of the first dynamic comparison scenario.

To better understand the results, statistical information about the traffic is shown in Table 1. Note that the lowest response time is during the fourth stream that was defined by highest traffic intensity. In additional the statistics confirm that the maximal initial response time relates to the network anomaly mentioned earlier, resulting in the slowest response time of 0.543ms

Table 1. Statistical analysis of the first dynamic comparison scenario.

Experiment stream descriptor	Packet generation intensity per second within the stream	Average response time [ms]	Minimum response time [ms]	Maximum response time [ms]
1st stream	100	0.16756	0.14600	0.38800
2nd stream	50	0.17329	0.15400	0.24700
3rd stream	500	0.15617	0.13300	0.54300
4th stream	1000	0.15055	0.13100	0.29200
Entire load	-	0.16189	0.13100	0.54300

Table 2. Distributions of the response times across the first dynamic comparison scenario.

Time difference with regards to the average response time [seconds]	Number of responses	Percentage [%]
0.001	16	0.40%
0.0001	78	1.95%
0.00005	3906	97.65%
-0.00005	0	0.00%
-0.0001	0	0.00%
-0.001	0	0.00%
Total	4000	100.00%

Enhanced analysis of the response time is presented in Table 2. The response time of the packets show that almost 98% of the traffic has been processed in 0.05ms time window from the average response time. This would suggest that the framework is managing the traffic in a consistent manner, even if the traffic quantity is variable.

4.1.2 Second cross dynamic comparison experiment

The second dynamic scenario was used to model the fluctuations in traffic intensity that often occur in real-world situations. The analysis is presented in Table 3.

Table 3. Statistical analysis of the second dynamic comparison scenario.

Experiment stream descriptor	Packet generation intensity per second within the stream	Average response time [ms]	Minimum response time [ms]	Maximum response time [ms]
1st stream	500	0.15161	0.13500	0.39800
2nd stream	1000	0.14777	0.13000	0.25600
3rd stream	50	0.16577	0.15200	0.31600
4th stream	100	0.15614	0.14200	0.25100
Entire load	-	0.15532	0.13000	0.39800

This provides further clarification to the first dynamic scenario. The lowest average response time was reached during the higher intensity of the traffic. Distribution of the response time window's management is provided in Table 4. The analysis shows that 99.4% of all traffic was processed within in 0.05ms. The table provides clarification on the consistency on the traffic processing.

Table 4. Distributions of the response times across the second dynamic comparison scenario.

Time difference with regards to the average response time [s]	Number of responses [responses]	Percentage [%]
0.001	3	0.08%
0.0001	21	0.53%
0.00005	3976	99.40%
-0.00005	0	0.00%
-0.0001	0	0.00%
-0.001	0	0.00%
Total	4000	100.00%

4.1.3 Third cross dynamic comparison experiment

The third scenario was designed to test the framework when subjected to significant increase and then decrease in traffic. The results of this analysis are shown in Table 5.

Table 5. Statistical analysis of the third dynamic comparison scenario.

Experiment stream descriptor	Packet generation intensity per second within the stream	Average response time [ms]	Minimum response time [ms]	Maximum response time [ms]
1st stream	50	0.16442	0.14900	0.37400
2nd stream	1000	0.15066	0.13500	0.26600
3rd stream	100	0.15748	0.14100	0.24200
4th stream	500	0.15172	0.14000	0.24200
Entire load	-	0.15607	0.13500	0.37400

The scenario also shows the lowest response time during the highest intensity of the traffic. Analysis of the distribution of the response time window is shown in Table 6. The table shows that the result of the distribution in almost 98% of the requests were processed in the 0.05ms window.

Table 6. Distributions of the response times across the third dynamic comparison scenario.

Time difference with regards to the average response time [s]	Number of responses [responses]	Percentage [%]
0.001	4	0.10%
0.0001	79	1.98%
0.00005	3917	97.93%
-0.00005	0	0.00%
-0.0001	0	0.00%
-0.001	0	0.00%
Total	4000	100.00%

4.2. Cross dynamic comparison evaluation

Each dynamic comparison scenario was conducted with the same quantity as well as intensity of the traffic. This particular set up of the dynamic scenarios allows further cross-referencing with previous dynamic scenarios conducted for the Enhanced DDoS-MS framework. Each of these scenarios contained multiple different stream intensities and the analysis focused on each stream as well as across the whole scenario.

The initial statistical analysis is presented in Table 7, where streams were separately analysed to provide insight regarding minimal, average and maximum response times. The table provides this information in order of the stream intensity. To make the compression clear the table shows the response time per intensity for each solution. The two-response times were deducted from each other to provide the response time difference in described cases. The difference is also calculated in terms of the percentage perspective. The results show that the average response time for one hundred packets per second is 0.565ms for the predecessor, and the newly new framework can handle the same quantity and intensity with the average response time 0.168ms. As shown in the table this means a time difference of 0.397ms, which represents 337 percent difference in comparison with the result achieved by the predecessor. This stream was taken as an example as it achieved the largest difference across comparable experiments. The results show that across these streams that the difference was in the milliseconds ranges, meaning in most cases an improvement of several hundred percent.

Table 7. Cross-correlation of results between in the Enhanced DDoS-MS and Cloud based DDoSAS

Intensity	Function	First scenario response time Enhanced DDoS-MS [ms]	First scenario response time Cloud based DDoSAS [ms]	Response time difference [ms]	Percentage difference in response time [%]
50	AVG	0.544	0.173	0.371	314%
50	MIN	0.415	0.154	0.261	269%
50	MAX	0.767	0.247	0.520	311%
100	AVG	0.565	0.168	0.397	337%
100	MIN	0.394	0.146	0.248	270%
100	MAX	0.830	0.388	0.442	214%
500	AVG	0.454	0.156	0.298	291%
500	MIN	0.337	0.133	0.204	253%
500	MAX	0.595	0.543	0.052	110%
1000	AVG	0.410	0.151	0.259	272%
1000	MIN	0.333	0.131	0.202	254%
1000	MAX	0.549	0.292	0.257	188%
All	**AVG**	**0.493**	**0.162**	**0.331**	**305%**
All	**MIN**	**0.370**	**0.141**	**0.229**	**262%**
All	**MAX**	**0.685**	**0.368**	**0.318**	**186%**

Higher level of visibility was achieved with similar analysis that included second and third comparable scenarios. It showed that the response time difference differs by 305 percent. Analysis showed that the minimal reached 262 percent and maximal response time had 186 percent difference. It suggests the while in general traffic is processed three time faster, the maximal response time are less than twice as fast. A possible explanation for this is that external factors are influencing some of the requests, which are then processed and delivered slower. This however is not happening to most of the traffic as the average response time results are in higher improvement rates.

The second and third dynamic comparable scenarios were analysed in the same manner. The observation that was identified within the first comparable scenario is visible also in these scenarios. The maximal response times have the lowest improvement; however, it is still faster than its predecessor.

4.3. Enhanced dimensions experiment

To provide further clarification on scalability, availability and verification, an additional experiment was conducted varying multiple input parameters. Here the quantity of the traffic was increased as well as the intensity of the traffic generated. The first comparison scenario served as a blueprint of the enhanced dimensions experiment. The intensity of the network packet generation was increased by five hundred percent. The same level of increase was applied to the quantity of traffic generated and sent towards the protected system and then processed through the framework. Therefore, the number of ICMP requests was raised to five thousand and the total traffic reached ten thousand within each stream.

The initial quantity was eight thousand packets, and thus when increased by a factor of five reached forty thousand packets. Moreover, while the maximal intensity of the traffic was initially one thousand packets generated per second it was increased to five thousand packets. Figure 3 graphs the results obtained.

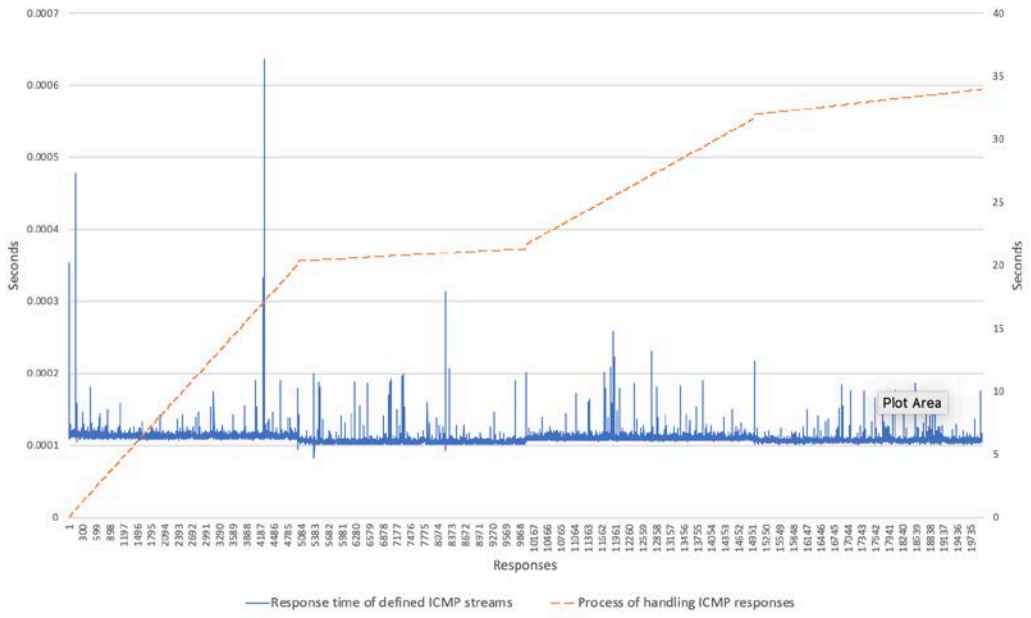

Figure 3. Results of the enhanced dimensions scenario.

The graph shows that there are several exceptions in the response times that significantly exceed the average. Table 8 provides the statistical analysis of this experimental scenario.

The lowest response time is 0.083ms, the highest response time is 0.386ms and the average response time of the experiment is 0.105ms. This experiment is a crucial in view of the previously identified observation that the framework achieves lowest response times during the highest intensities. In this situation the lowest response time is not achieved during the highest intensity of the five thousand packets per second that was generated in the second stream. The analyses of the

distribution of the response times in the 0.05ms window from the average response time is shown in Table 9.

Table 8. Statistical analysis of the enhanced dimensions scenario.

Experiment stream descriptor	Packet generation intensity per second within the stream	Average response time [ms]	Minimum response time [ms]	Maximum response time [ms]
1st stream	250	0.10547	0.09800	0.36800
2nd stream	5000	0.10567	0.10000	0.20200
3rd stream	500	0.10500	0.09700	0.20100
4th stream	2500	0.10512	0.08300	0.20100
Entire load	-	0.10533	0.08300	0.36800

Table 9. Distributions of the response times across the enhanced dimensions scenario.

Time difference with regards to the average response time [s]	Number of responses [responses]	Percentage [%]
0.001	4	0.02%
0.0001	41	0.21%
0.00005	19955	99.78%
-0.00005	0	0.00%
-0.0001	0	0.00%
-0.001	0	0.00%
SUM:	20000	100.00%

The distribution table suggests that the rate of responses served within 0.05ms is consistent with the previous comparisons scenarios, as it is kept at the same levels. More precisely it achieved the highest percentage of the packets processed within this timeframe in comparison with all the previous comparison scenarios.

5. Concluding Discussions

Increasingly businesses are adopting cloud computing facilities that rely on non-private computer networks. This raises a number of security concerns that must be addressed. This paper focuses on one aspect of these concerns, namely denial of service attacks. The paper presents the Cloud based Distributed Denial of Service Alleviation System framework (DDoSAS) that has been implemented using Amazon Web Services. This real-life environment facilitated experiments that highlighted an efficiency, in terms of end-to-end latency, substantially better than obtained with previous frameworks and faster response times during higher traffic intensity. Moreover, the new framework can handle five thousand requests per second and still achieve an average response time of 0.105 millisecond. Observation spanning across all the conducted scenarios show that above 97 percent of all requests are handled within 0.05 ms from the average response time, which suggests that the framework provides consistency regardless of traffic load, dynamic scenario or intensity.

References

[1] Cloud Computing Market Projected To Reach $411B By 2020 Available online: https://www.forbes.com/sites/louiscolumbus/2017/10/18/cloud-computing-market-projected-to-reach-411b-by-2020/#d01322278f29.

[2] Khalid Al-Begain, Michal Zak, Wael Alosaimi, and Charles Turyagyenda, "Security of the Cloud", in Emerging Research in Cloud Distributed Computing Systems, Bagchi, S. (ed.). Hershey, PA: IGI Global, pp. 363-404. ISBN 978-1-4666-8213-9.

[3] The NIST Definition of Cloud Computing, Available: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf (accessed on 25 Oct 2019).

[4] Mohammad Moradi and Mohammad Keyvanpour, "CAPTCHA and its Alternatives: A Review. " Security Comm. Networks, 8: 2135– 2156. doi: 10.1002/sec.1157. Available: https://onlinelibrary.wiley.com/doi/pdf/10.1002/sec.1157 (accessed on 24 Nov 2019).

[5] Gartner Says Worldwide IaaS Public Cloud Services Market Grew 31.3% in 2018, Available online: https://www.gartner.com/en/newsroom/press-releases/2019-07-29-gartner-says-worldwide-iaas-public-cloud-services-market-grew-31point3-percent-in-2018 (accessed on 28 Nov 2019).