Improving network intrusion detection system performance through quality of service configuration and parallel technology

Bul'ajoul, W. , James, A. and Pannu, M.

Improving Network Intrusion Detection System Performance through Quality of ServiceConfiguration and Parallel Technology

Waleed Bul'ajoul
Faculty of Engineering and Computing
Coventry University
Coventry, UK
bulajouw@coventry.ac.uk

Anne James
Faculty of Engineering and Computing
Coventry University
Coventry, UK
csx118@coventry.ac.uk

Mandeep Pannu
Department of Computer Science
Kwantlen Polytechnic University
Surrey,
British Columbia,
Canada

Abstract.

This paper outlines an innovative software development that utilizes Quality of Service (QoS) and parallel technologies in Cisco Catalyst Switches to increase the analytical performance of a Network Intrusion Detection System (NIDS) when deployed in high-speed networks. We have designed a real network to present experiments that use a Snort NIDS to demonstrate the weaknesses of NIDSs, such as inability to process multiple packets and propensity to drop packets in heavy traffic and high-speed networks without analysing them. We tested Snort's analysis performance, gauging the number of packets sent, analysed, dropped, filtered, injected, and outstanding. We suggest using QoS configuration technologies in a Cisco Catalyst 3560 Series Switch and parallel Snort NIDSs to improve NIDS performance and to reduce the number of dropped packets. Our results show that our novel configuration improves performance.

Keywords: network security; intrusion detection system; parallel; Quality of Service

I. INTRODUCTION

In order to provide new developments and the highest-quality services, companies implement the latest technologies in their infrastructure. A company's network plays a vital role in its business projects; it can achieve success in its business career by keeping its computer network up-to-date with the latest software and security techniques. Reliability and safety are the major concerns in enabling a company to achieve success and boost its progress. However, these networks can also be considered a major risk in any business project.

Security issues have increased as technology has advanced. Fuchsberger [1] reported that, according to a survey conducted by Federal Bureau of investigation and Crime Scene of investigation (FBI/CSI), viruses are behind many attacks on business networks. Moreover, denial of service (DoS) attacks and unauthorized user access (which can be initiated from external or internal LAN sources) have also increased dramatically.

It is also noticeable that nowadays there are powerful intrusion tools available, allowing hackers to attack networks even if they know little of the software. Attackers can now use several tools simultaneously to achieve an objective. However, 9th Annual Worldwide Infrastructure Security Report and ATLAS data 2013[2] report said the number of DDoS attack has grown significantly, nearly doubling on a year-to-year basis between 2006 and 2010. The size peaks of attacks in 2013 has been increased by over 200 percent form previous year, with the largest reported attack at 309 Gbps, and with multiple respondents reporting attacks larger than 100 Gbps – the previous largest reported attack size [2]. Additionally, in 2013 ATLAS observed more than 8x the number of attacks over 20 Gbps tracked as compared to 2012 [2].

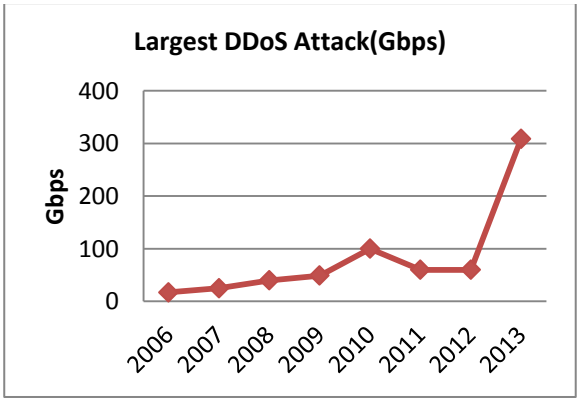

Figure 1: Largest DDoS Attack Reported by Arbor Networks [2].

Therefore, security products such as firewalls, vulnerability assessment tools such as antivirus programs and Intrusion Detection Systems (IDSs), security appliances such as the Cisco ASA 5500 series, etc., may reduce the risk of attacks. However, even these measures are not 100 percent effective in protecting networks. As a result, multiple products are combined together to strengthen the network security seals.

1. Firewall

In order to secure a corporate network or sub-network, network traffic is usually filtered according to criteria such as origin, destination, protocol or service, typically through dedicated routers called firewalls. Firewalls are a common security defence and nowadays are treated as an integral part of every network. They are now part of every network gateway, preventing external intruders from gaining access to local or private networks [3].

A firewall may be software or hardware; its functionality is based on filtering mechanisms specified by a set of rules,

known as a policy, which can protect a system from flooding attacks. The fundamental function of a firewall is to sort packets according to allow/deny rules, based on header-filed information [3].

The disadvantage of firewalls is that they cannot fully protect an internal network; they are unable to stop internal attacks [4, 5]. For example, malicious and unwanted web traffic can go through a firewall to strike and damage a protected computer system without a hitch.

A firewall is just a set of rules such as to allow or deny protocols, ports or an IP address [3]. Today's Denial of Service (DoS) attacks are too complex for firewalls because it they cannot distinguish good traffic from DoS attack traffic [3].

2. Antivirus programs

Computer viruses are programs which cause computer failure and damage computer data. Especially in a network environment, a computer virus poses an immeasurable threat and can be very destructive [5]. Antivirus programs are software that can be installed onto a computer in order to detect, prevent and make decisions regarding whether to quarantine or delete malicious programs such as malware, worms or viruses.

Although antivirus programs monitor the integrity of data files against illegal modifications, they are unable to block unwanted network traffic intended to damage the network. Anti-threat software is installed only at explicit points of the servers, such as the interface between the network segment to be protected and outside environments [6].

3. Intrusion detection systems (IDSs)

The firewall is an interesting technique and provides the benefit of added security to strengthen a network when used in conjunction with an Intrusion Detection System. IDS technologies detect and react to unauthorised access to network systems, providing real-time monitoring of network traffic [7].

IDSs can be software- or hardware-based, or can be a combination of both. Hardware-based IDSs are effective for large organizations and companies, but are very expensive. However, software-based IDSs running on the same devices or servers can identify and deal with attacks generated from inside or from outside the network, and can also protect the security policies of that network and their internal threats. It is an interesting and beneficial way to implement a technique that can be used to multitask and provide extra security techniques to strengthen your network along with an IDS by deploying a firewall or any other technique [7, 8 and 9].

Intrusion detection (ID) is one of the most tested and reliable technologies to monitor incoming and outgoing network traffic to identify unauthorized usage and mishandling of computer system networks [10]. In addition, ID identifies the activity of malicious attackers. It is critical to implement ID in computer networks that have high traffic and high-speed connectivity [11, 12].

4. Cisco ASA 5500

The ASA 5500 series is a range of essential Cisco products that can secure an organisation's network from end to end, and comes in different sizes. The scalability of the ASA 5500 series provides a vast range of effective security solutions. It has therefore been a top choice for network designers for its high performance, and many vendors are now trying to produce security appliances that can protect networks [13].

This paper which builds on our previous work [12], describes research which uses Snort IDS, the most popular Network Intrusion Detection System (NIDS) software. Snort can be installed in any machine that runs on different operating systems such as Windows, Linux, etc. Snort was introduced as a lightweight IDS and has developed significantly in the last 10 years.

We conducted some experiments to test Snort NIDS analysis performance under heavy and high-speed traffic. We also demonstrated that Snort NIDS's performance can increase the number of analysed packets and decrease the number of dropped packets using alternative technologies such as a Quality of Service (QoS) configuration and parallel NIDS technology. In this paper, we will outline some concepts related to Network Intrusion Detection and Prevention System techniques used to mitigate security problems and risks.

The remaining part of this paper is organized as follows: Section II gives a background about security stages and IDS technologies. Section III offers a detailed discussion of IDS methodologies, while Section IV covers the major components of NIDS. Section V explains our experimental design and implementation; covering details about experiment devices and tools. Section VI presents the metrics approach used in the evaluation of our experiments. Section VII presents the results of the experiments and the evaluation. Section VIII gives an overview related work. Finally Section IX concludes the paper and suggests recommendations and further work.

II. BACKGROUND

Security is a major concern in every aspect of our daily life. New methods and equipment are always being devised to ensure protection; similarly, computer networks continue to face many threats [2].

There are usually three stages to achieving security in computer system networks: prevention, detection and correction [14]. Prevention stops attacks before they enter system, Detection catch the attacks after they have entered and then Correction rectifies problems, which could be detected attacks or mistakenly prevented non-attacks.

Prevention is the ideal solution, as compared to detection and correction, but it is impossible to prevent 100 percent of attacks [15, 10]. Moreover, detection techniques provide more accurate results in deterring malicious attackers than correction techniques.

The correction technique is used to protect computer systems. Along with the prevention technique, it actively works to block intrusions, but it can continue to battle a successful intrusion [12]. Nevertheless, a number of successful attacks can be controlled using the prevention technique if an attack is detected at the interim stage. This is difficult, because some successful attacks can get through the prevention system [16, 17]. It is a matter of a system being attacked, compromised, and consequently malfunctioning. Here we need an interim stage, such as the detection phase, which should be activated by intrusions. The detection method is preferred as it minimizes network costs and fills in the gap between correction and prevention mechanisms.

1. Intrusion prevention system technology (IPSs)

The IPS is the advanced version of the ID scheme comprised of a router, a firewall, and the filtering of a network layer. This technology helps to make a network more secure. IPSs filter traffic, determining whether to allow it to pass or to block it if it contains malicious viruses. They are also useful in combating flood-type attacks, including Distributed Denial of Service (DDoS) and DoS [1].

The weaknesses of IDS (such as overhead on monitored system , performance in high speed network, false positive alert, protection against new attacks and others) resulted in research that has lead to the development of IPs [12, 18, 19], which are in-line, active devices that can drop packets or stop connections that are malicious before they reach their targets.

IPs need to be configured properly to achieve the desired output. There are three classifications of IP:

- those that can stop an attack themselves;
- those that can alter attack contents; and
- those that can change the security environment [18, 19].

2. IDS technology

A firewall is not useful for attacks launched within a network, as it cannot detect the intrusion; it just drops the packets. For this reason, a more powerful tool is required. IDSs are more enhanced and better-developed security tools than firewalls. A useful analogy for an IDS is to think of it as a burglar alarm [20].

This technology detects unauthorized intrusions of the network or host that could damage or compromise network devices. If it finds any unusual activity, it sends the suspicious activity to the monitoring centre. The analysis and translation of network events and traffic is performed by this technology. It also has access to a database containing the signatures and profiles of known attackers or malicious behaviours, where it can compare any kind of suspicious activity to the one stored in their database, which helps in reacting to these activities, including shutting down a connection, alerting who to an attack, and logging activity for further investigation [3, 21].

IDSs are still unable to control all threats and malicious activities [3, 12, 16, 18]. To overcome the design and implementation difficulties, novel IDS outcomes have been obtained from multiple characteristics of advanced computer networks:

- processing in real time;
- high speeds and high loads;
- increasing difficulties for defenders; and
- reducing difficulties for attackers.

3. Intrusion detection and prevention system (IDPS) technologies

Intrusion detection and prevention system (IDPS) technologies may be software, hardware, or a combination of both. Snort, a tool that can be used in both IDSs and IPSs, and can perform packet sniffing or real-time analysis on local area networks [22].

Some of the existing types of IDSs are: network-based (NIDS); host-based (HIDS); graph-based IDS (GrIDS); and hybrid-based.

A) Network-based IDSs

A Network Intrusion Detection System (NIDS) is a common technique used to analyse traffic at all Open Systems Interconnection (OPI) layers by detecting the presence of normal traffic or suspicious activities [23].

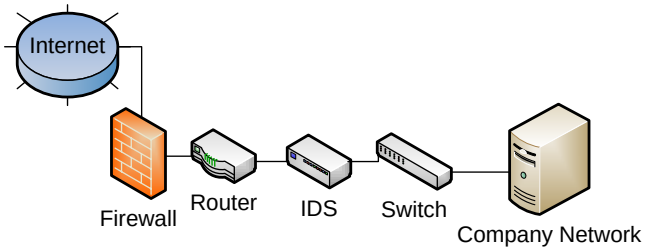

Figure 2: An Example of Network-Based IDS

A disadvantage of NIDS is that it is usually unable to execute entire network packets, which results in incomplete analyses and therefore considerable delays in high-speed and high-load environments [8, 12]. NIDS itself is affected by DoS and DDoS attacks, similar to those made against IP gateways. In addition, NIDS is unable to inspect encrypted network traffic (packets) due to being placed in the middle of a network connection. Therefore, an NIDS only works in network-based environments, not at the end points of a network [16, 17].

B) Host-based IDSs (HIDSs)

A Host Intrusion Detection System (HIDS) is a software agent that can be installed in a particular computer in order to monitor and analyse events on that particular host to detect any suspicious behaviour [24].

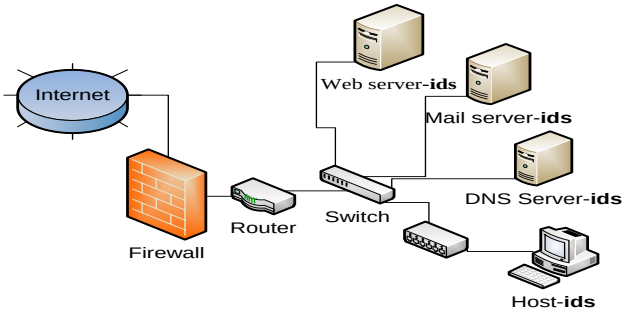

Figure 3: An Example of Host-Based IDS

Advantages of HIDSs include:

- They are capable of integrating code analysis, monitoring system calls, detecting buffer overflows, privileging misuse, privileging abuse, file systems, library lists, applications, system configurations, system log analyses, and many others [19]. These things can be done by HIDSs because they are designed to operate with a specific host and with respect to applications such as web servers, database servers, file servers, mail servers, and DNS servers.
- They are often integrated into server software and can be relatively easily implemented to communicate with other network components and operating systems.
- They can inspect encrypted traffic because they can analyze packets at the application ends [12].

Disadvantages of HIDSs include:

- HIDSs consume computer system resources that are needed for services.
- HIDSs may conflict with existing security policies (such as firewalls) and operating systems.

- It is difficult for HIDSs to analyse intrusion attempts on multiple computers.
- HIDSs can be very difficult to maintain in large networks with different operating systems and configurations.
- HIDSs can be disabled by attackers once a system is compromised.
- HIDSs require many hosts to reboot after a complete installation or an update [16].
- Many essential servers cannot support this operation.

C) Graph-based IDSs (GrIDSs).

Graph-based IDSs (GrIDSs) are designed to protect computer networks from large-scale malicious attacks, which severely affect computer networks. Network traffic and computers are linked through GrIDs.

The advantages of GrIDSs are as follows:

- it can gather data about computer activity and it helps to recognize comprehensive automated or coordinated attacks in real time;
- it allows network systems to state and implement policies specifying which users are permitted to utilise the particular services of an individual host or group of hosts;
- it determines and reports violations of the stated and recommended policy [25]; and
- UNIX hosts are used to run GrIDSs, which are in turn connected to the Internet [26].

Assumptions made in this kind of system include:

- The inclusion of related networks within a single organisation that has an independent infrastructure and sovereign departments. Despite this, it is even more difficult to picture how this system would work to gain insight into the working of the GrIDSs system in a modern and innovative enterprise environment.
- No single component of the network is actively hostile, which is a problem because the IDS must be redesigned to operate in non-hostile situations.

D) Hybrid-based IDSs

A Hybrid Intrusion Detection System is technology that combines two or more IDSs in order to analyse, detect, and match any suspicious behaviour or signature malicious code that attempts to attack a network [11].

III. Intrusion Detection System Methodology

Most existing NIDS utilize either misuse detection or non-regular detection. The technique of misuse detection is employed to find known intrusions and/or a pattern of signatures. Due to its reliance on signatures, its detection speed is quite fast and has a low false positive rate.

The NIDS methodology is divided into the following four categories:

1. Misuse / signature-based detection

This type of detection system uses known signatures of malicious codes, which are stored in an IDS database. Well-known patterns of attacks result from the use of malicious codes and known software vulnerabilities.

This kind of detection system is highly efficient for use in a small NIDS or Host-based IDS. The major drawback of such a system is that its database must be regularly updated, resulting in an ever-increasing database that must include as many available signatures as possible [12, 18, 27].

2. Anomaly/statistical detection

Anomaly/statistical detection is a comparison-based method which compares any activity to the profile for all possible learned actives through statistical data, facts and figures. There are two types of profile, fixed and dynamic. A fixed profile is the most efficient as compared to other schemes, because it terminates the occurrence of any unusual behaviour and it classifies the behaviour as anomalous. A dynamic profile cannot be created without an existing fixed profile; once the dynamic profile has been created, it allows the attacker to observe and alter his or her behaviour in long-term activities [12, 18, 27].

3. Protocol analysis detection

This IDPS technique depends on the behaviour of the protocols. It observes the protocol behaviour and then compares it to those stored in its protocol behaviour database. It detects anomalies in the packet on the head part of the protocol. This technique is quite effective, but can be easily avoided by attackers working inside the protocol limitations [12, 20, 27].

4. Hybrid methodologies

This system is the combination of more than one technique. It combines two or more intrusion detection systems methodology in order to analys, detect and match any suspicious behaviour and signature malicious code that attempt to attack network. However, instead of identifying a single intrusion, it detects multiple intrusions, thereby providing relatively better results as compared to other methods with greater strength [6, 12, 27].

IV. The Components of Network Intrusion Detection Systems

The functional components of IDSs are: events management, a data source, an analysis engine, and a response manager [18, 27].

1. Events management

Events management gathers information on events to and from the monitoring system (see Figure 4).

2. Data source

The data source is the event generator, which is classified into the following four categories [27]:

- application-based monitors;
- host-based monitors;
- target-based monitors; and
- network-based monitors.

The data source stores multiple events recorded by event management.

3. Analysis engine

The analysis engine collects data from the data source in order to analyse and determine whether the data is free of policy violations or other attacks. This engine can utilize anomaly/statistical detection, misuse/signature-based detection, or both [27]. The analysis engine processes events and transmits alerts.

4. Response manager

The response manager neutralizes an attack once it is detected. Any IDS, including HIDS or NIDS, can utilize this

analysis system. The response manager responds to events and stops intrusions [27].

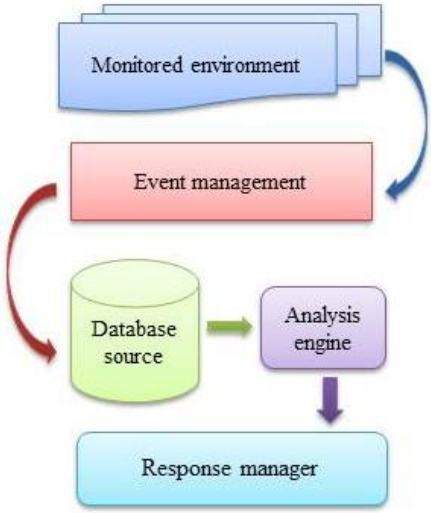

Figure 4: General Architecture for NIDSs

V. Experiment Design and Implementation

A) *Experiment design*

A real network has served as a model for the purpose of analysis and data acquisition. The following factors have been taken in to consideration for this design:

- manual data access;
- configuration/setup;
- cost-free tools;
- availability of tutorials; and
- support related to work

In this study, we used several tools, including both software and hardware, to meet the objectives outlined in the introduction: Snort 2.9.4.6, which was issued in April 2013, was introduced as NIDSs software; a WinPcap tool to capture packets on Windows 7 and 8 operating systems; a NetScanPro tool to manage a certain type of traffic in the network; a Packet Generator tool to generate/send network traffic of different values and speeds per ms; a Cisco Catalyst 3560 Series Switch, which supports QoS configuration; and a computer system consisting of a minimum of four PCs with VMware Virtual. Figure 5 shows the network design for the experiment.

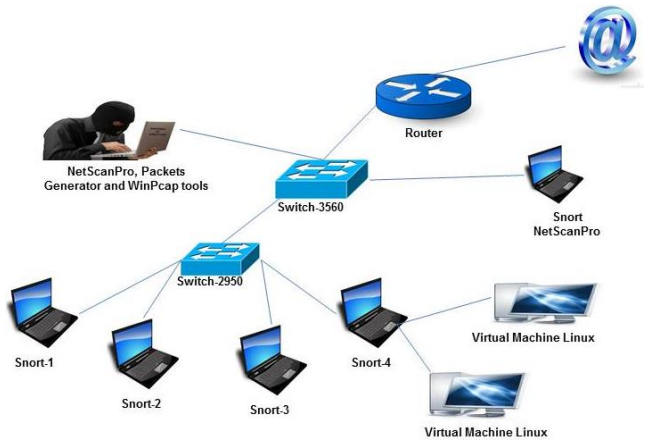

Figure 5: Experiment Network Design

B) *Snort component functions*

Snort is one of the easy to use and popular IDSs [28]. It is accessible free of cost and ranked among the top systems with the best features available nowadays. It was released as an open-source NIDS based on a rule-based IDS, which stored information in text files that can be modified by a text editor. Rules are grouped into categories, and the rules belonging to each category are stored as information in separate files, which are then integrated into the main configuration file named "snort.conf". The data is captured in terms of the described rules, which are read at the initialization of the Snort and comprise the internal data structure [12, 20, 28].

A Snort-based IDS consists of the following major components:

- a packet decoder;
- pre-processors;
- a detection engine;
- a logging and alerting system; and
- output modules.

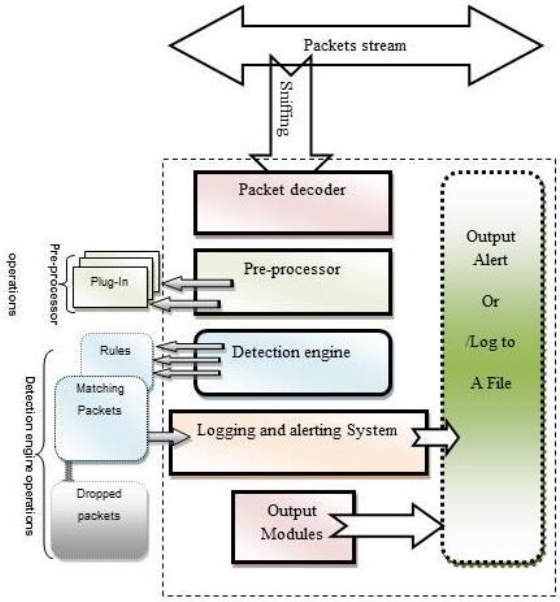

Figure 6: Snort Architecture [16]

The basic structure is represented in Figure 6. When a packet arrives at the network, Snort listens and captures packets. In the beginning, the packet decoder receives packets from multiple network interfaces such as Point-to-Point Protocol (PPP) or Ethernet and Serial-Line-Internet-Protocol (SLIP), then organises such packets for pre-processing using a detection engine [12, 28].

The pre-processor filters quickly organise and modify the data packets before transferring them to a detection engine, such as multiple UDP and/or TCP packets and port numbers [12, 20]. The pre-processor is very important part of any type of IDS to prepare data packets to be analysed against rules in the detection engine [28]. The detection engine performs three main tasks: sniffing, analysis and detection. The detection engine can be used to perform any one of the three tasks.

The detection engine is time-critical and the most important part of the Snort. It utilizes different processing times based on the length of the packet, the specifications of the system, and the number of rules defined in the system.

Snort drops few packets because it runs in real time if the operation is in NIDS mode, as with heavy and high-volume traffic [12, 20].

This technique employs the Snort rule to detect intrusive actions to be presented in the data packet. In the detection mode, Snort rule is capable of reading chains (internal data structures), which have to be matched against all packets. If a packet does not match any rule, it will be dropped; otherwise appropriate action is taken [12, 20, 28].

Logging and alerts depend on the nature of what is detected inside the packets. If any suspicious activity is found inside a packet, the packet usually logs the malicious activity and/or generates an alert. Logs are usually stored in simple text-based files such as tcp-dump-style, etc. [12, 20].

Output modules (plug-ins) are capable of performing multiple operations depending on the results generated by the logging and alerting system of Snort. In general, output modules control the form of outcome produced by the logging and alerting system [12, 20].

C) Cisco Catalyst 3560 Series switches

This category belongs to layer 2 and 3 switches. It provides support for IP-based software, for example Rate limiting, Access Control Lists (ACLs), QoS. IPv6, and advanced routing protocols. Policy and class enterprise features are supported by IP service software. Despite a packet's size and content, this switch provides the best effort services for each packet of network traffic. The packets are sent with no surety of delay bounds, reliability, or throughput [13].

D) Quality of services (QoS)

A QoS technique permits the control of traffic over a network and guarantees the throughput of traffic applications in terms of time scale. QoS is the performance of the network traffic over several technologies, including Asynchronous Transfer Mode (ATM), 802.1 networks, IP-routed networks, Frame relay, and Synchronous Optical Network (SONET) as seen from the user's perspective. Furthermore, QoS can use congestion avoidance and management techniques along with configuration of network traffic, and prioritizes traffic based on its importance [29].

QoS features can be classified into the following functions:

- classification and marking;
- policing;
- congestion management; and
- congestion avoidance.

The features of QoS provide better and more reliable network services through the following features:

- support for dedicated bandwidth;
- improved loss characteristics;
- management and avoidance of network congestion;
- shaping network traffic; and
- Setting traffic priorities across the network.

VI. EXPERIMENT PERFORMANCE METRICS APPROACH

The performance metrics are used in the experiments to measure the ability of a NIDS to perform a particular task and to fit within the performance constraints. These metrics measure and evaluate the parameters that impact NIDS performance. The measure metrics of experiments were used are defined as the following below:

1. Packets generators

The performance of TCP, UDP, and ICMP protocols was measured when running over the IPv4 header. WinPcap and Packets Generator tool were used to vary the type of traffic in term of IP header protocol (TCP, UDP and ICMP), speed, the number of packets and packet size.

2. Timing statistics

The Snort processor time includes total seconds and packets as well as packets processing rates.

3. Packets I/O totals

These are the percentages of the total packets processed by Snort. The specific metrics used are shown in the table 1.

Table 1. Snort performance metrics

Performance metrics	Description
Packets received	The number of packets received by Snort.
Packets analysed	The percentage of packets analysed from the total packets received.
Packets dropped	The percentage of packets dropped from the total packets received.
Packets filtered	Packets are not shown for snort.
Packets outstanding	The number of the packets buffered waiting processing /or not processed.
Packets injected	Injected packets are the result of active response, which can be configured for inline or passive modes.

4. Protocol statistics

All traffic for all protocols decoded by Snort are summarized in the breakdown section includes categorises such as Eth (Ethernet interfaces); VLAN; IP4; Frag (Fragmented packages); ICMP; UDP; TCP and others.

5. Snort-NIDS throughput

This metric defines the level of traffic up to which the NIDS performs without dropping any packets. These metrics are affected by the use of QoS configuration and parallel technology.

VII. EXPERIMENT RESULTS AND EVALUATION

Snort is capable of performing real-time traffic analysis and packets logging on the network. It is a multimode packet sniffing, analysis and detecting tool. It can perform network traffic analysis and content searching/matching in both real-time and for forensic post-processing [22, 28]. However, our work has focused on the Snort capability as a Network based-IDS (NIDS). NIDS mode enables Snort to analyse the network traffic against a set of defined rules in order to detect intrusion threats. In our experiments, described later, we concentrate on the analysis function of the detection engine.

The purpose of the experiments is:

1) To show how Snort-NIDS performance under (a) high speed traffic (experiment 1), (b) heavy traffic (experiment 2) and (c) large data traffic (experiment 3).
2) To show how QoS configuration, which offers queue technology improve performance of Snort NIDS (experiment 4).

3) To show how parallel technology and QoS improve performance of Snort NIDS (experiment 5).

A) Experiment 1. Snort-NIDS reactions to high-speed network traffic

We used NetScanPro tools to manage IP traffic in the network and the packet generator tool to send a number of IP packets in different speeds per ms. We sent 13,000 1kb packets at different time intervals. Table 2 and Figures 7, 8 and 9 show the results of our experiments.

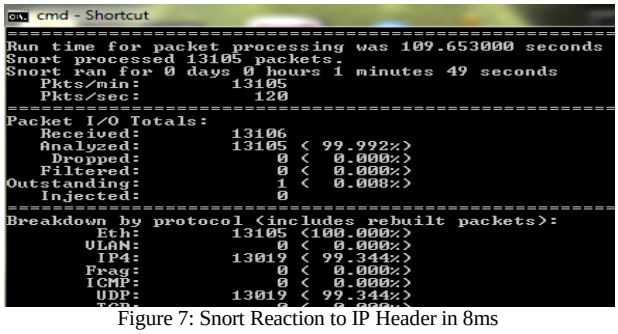

Figure 7: Snort Reaction to IP Header in 8ms

Figure 8: Snort Reaction to IP Header in 4 ms

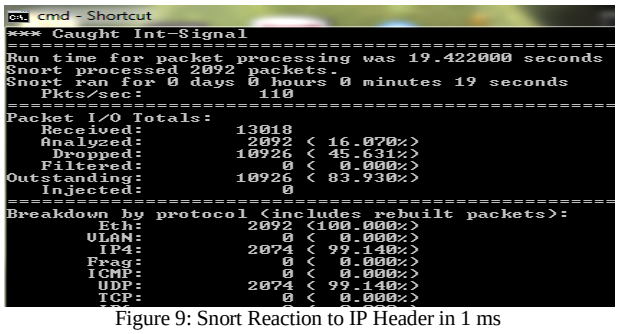

Figure 9: Snort Reaction to IP Header in 1 ms

Table 2. Same Value (Number of Packets) and Different Speeds

Packets sent (13,000)	8ms interval	4ms interval	1ms interval
Packets received	100%	100%	100%
Packets analysed	99.992%	62.165%	16.070%
Packets dropped	0.00%	27.449%	45.631%
Packets filtered	0.00%	0.00%	0.00%
Packets outstanding	0.008%	37.835%	83.930%
Packets injected	0.00%	0.00%	0.00%

As demonstrated in the results shown in Figures 7, 8, and 10, all the packets that were sent reached the wire. Snort analysed 99.992 percent of the packets in incoming traffic when packets were transmitted in 8ms intervals (see Figure 7), but when the speed of transmission was increased to 4ms, Snort started dropping packets, analysing only 62.168 percent and dropping more than 22 percent of the total packets received (see Figure 8). When the speed of transmission interval was increased to 1ms, Snort dropped more than 45 percent of packets (see Figure 9). Our experiment demonstrated that Snort analysis performance was decreased as the traffic speed limit increased.

B) Experiment 2. Snort-NIDS reactions to heavy-traffic networks.

Here, the transmission rate of packets was kept to the same speed (1ms intervals) to obtain a fair analysis of different values (each packet carried 1kb). We sent 100, 500 and 1000 packets batches at 1ms intervals.

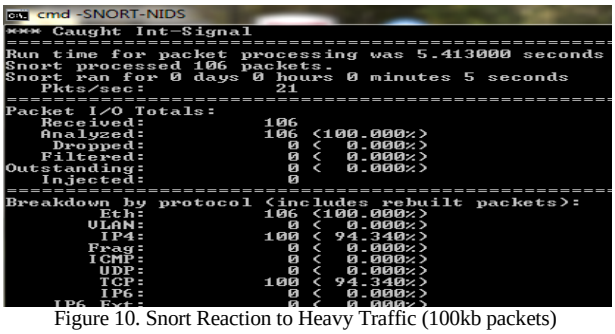

Figure 10. Snort Reaction to Heavy Traffic (100kb packets)

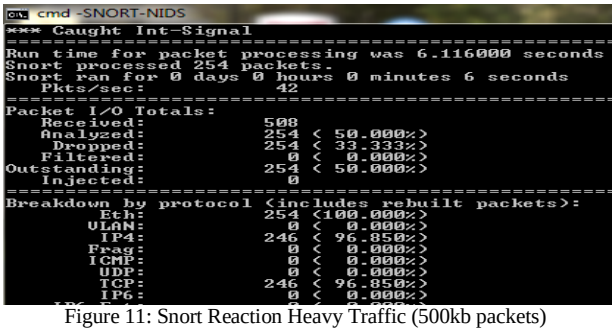

Figure 11: Snort Reaction Heavy Traffic (500kb packets)

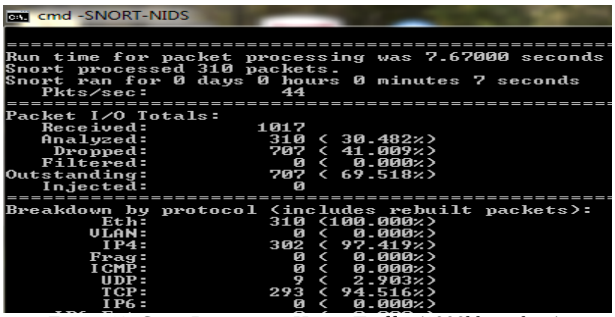

Figure 12: Snort Reaction to Heavy Traffic (1000kb packets)

Table 3. Same Speed Limit and Different Values

Traffic speed(1ms)	100kb	500kb	1000kb
Packets received	100%	100%	100%
Packets analysed	100%	50.000%	30.482%
Packets dropped	0.00%	33.333%	41.009%
Packets filtered	0.00%	0.00%	0.00%
Packets outstanding	0.00%	50.000%	69.518%
Packets injected	0.00%	0.00%	0.00%

As demonstrated by the results shown in Figures 10, 11, and 12, all the packets that were sent reached the wire. In Figure 10, when we sent 100 packets, Snort analysed 100% of the total packets that it received. As the number of packets

increased to 500 and 1000, Snort started dropping packets (see Figures 11 and 12). Our experiment shows that as the number of packets increases, more packets are dropped.

C) Experiment 3. Snort-NIDS reactions to large (length) packets

For this experiment, the number of packets was kept to the same value (13,000) and the same speed (1ms) to obtain a fair analysis of different sizes (lengths) of packets. We increased the size of each packet sent to 1 byte, 400 bytes, and 800 bytes.

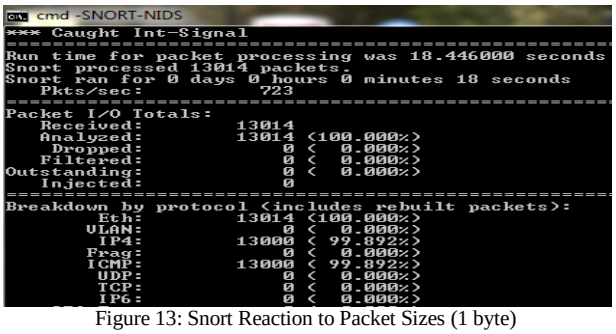

Figure 13: Snort Reaction to Packet Sizes (1 byte)

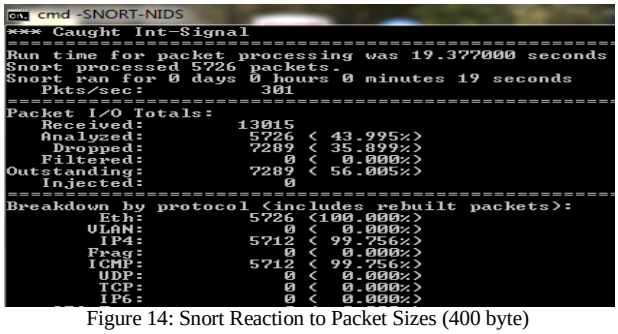

Figure 14: Snort Reaction to Packet Sizes (400 byte)

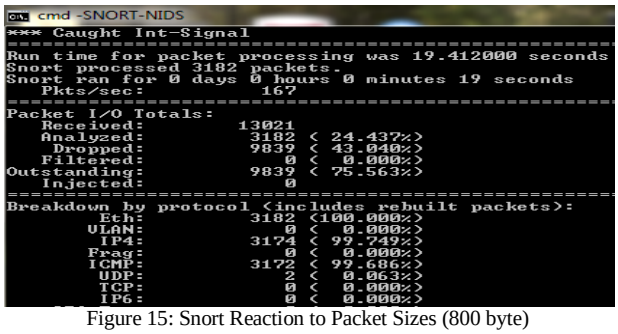

Figure 15: Snort Reaction to Packet Sizes (800 byte)

Table 4. Same Speed and Value but Different Size

Packets sent (13,000kbp1ms)	1 byte	400 bytes	800 bytes
Packets received	100%	100%	100%
Packets analysed	100%	43.995%	24.437%
Packets dropped	0.00%	35.899%	43.040%
Packets filtered	0.00%	0.00%	0.00%
Packets outstanding	0.00%	56.005%	75.563%
Packets injected	0.00%	0.00%	0.00%

As shown in Figures 13, 14 and 15, when we sent 13,000 packets in 1 ms, each packet carrying 1 byte, Snort analysed 100 percent of the total packets received (see Figure 13). As the size (length) of the packets was increased to 400 bytes, Snort dropped more than 35 percent of them (see Figure 14), and when the packet size was increased to 800 bytes, Snort accordingly dropped more (See Figure 16). Our experiment demonstrated that more packets will be dropped as packet size increases.

D) Experiment 4. Snort-NIDS using QoS configuration technology in high-speed traffic

Critical analyses were done for experiments 1, 2 and 3 (see Figures 16, 17 and 18 respectively). It was found that Snort performance analysis throughput was affected by high-speed and heavy traffic, and more packets were dropped as the number and size of packets and the speed of traffic increased. Because Snort has a limited time to process and analyse any traffic successfully if a network's traffic speed limit is higher than Snort's limit, Snort will drop packets.

To solve this problem, we used a Cisco Catalyst 3560 Series switch, which supports QoS configuration, to load the traffic into a number of interfaces equally and divide traffic into streams in order to analyse each portion of traffic individually to determine whether it was free of malicious codes. We configured Snort and QoS to reorder and control traffic speed as a Snort effort, similar to processor time and load traffic.

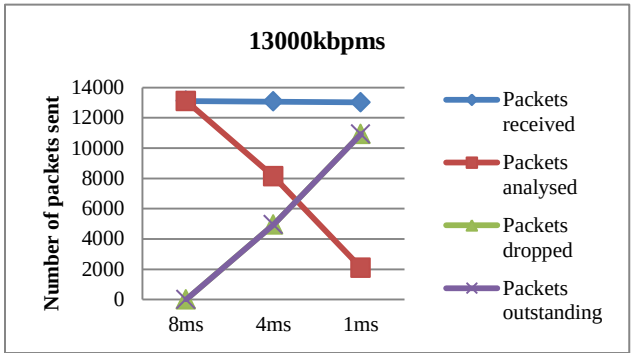

Figure 16: Snort Reactions to IP Headers with Increasing Traffic Speeds

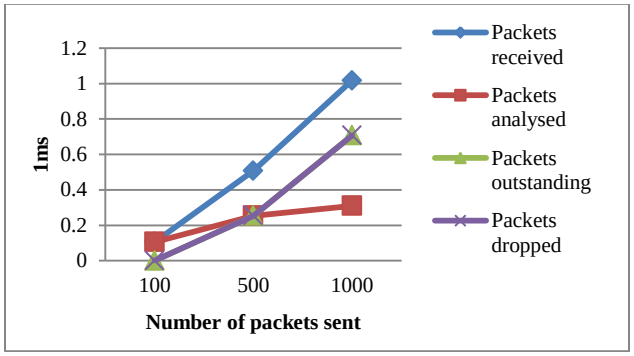

Figure 17: Snort Reactions to IP Headers with Increasing Traffic Values

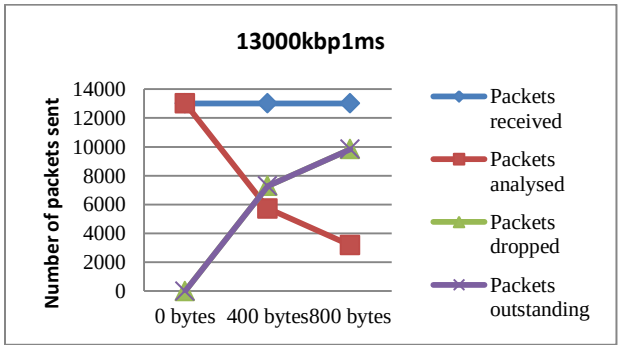

Figure 18: Snort Reactions to IP Headers with Increasing Packet Sizes

By default, Cisco switches work in layer 2, the Data Link layer and use the Class of Services (CoS) value [13, 29] (see Figure 19). In this layer commands support QoS configuration. To implement QoS based on the DiffServ architecture, which specifies that each packet be classified upon entry into the network and adjusted for different traffic speeds, we changed/classified the switch frame to the default, working from layer 2 to layer 3 by mapping the traffic values from CoS to Differentiated Services Code Point (DSCP) values.

To give preliminary treatment to packets with the same class information and different treatment to packets with different class information, a class map and policy map functions can be used to classify traffic inside the switch [29]. Classification is the process of distinguishing one kind of traffic from another by examining fields in the packet. Classification occurs on the physical interface or on a per-port, per-VLAN basis.

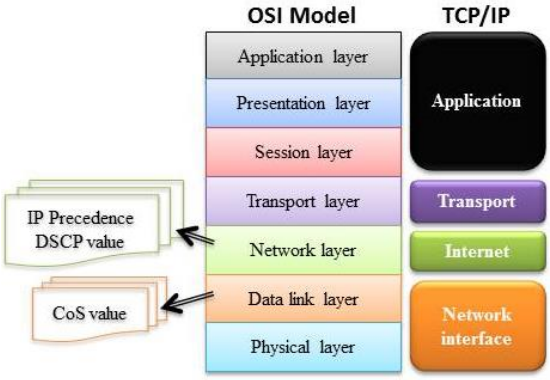

Figure 19:. Place for CoSand DSCP Values

CoS value: from 0 to 7, DSCP value: from 0 to 63

Policing involves creating a policy that specifies the bandwidth limits for the traffic and applies it to the interface. Policing can be applied to a packet per direction and can occur on the ingress and egress interfaces.

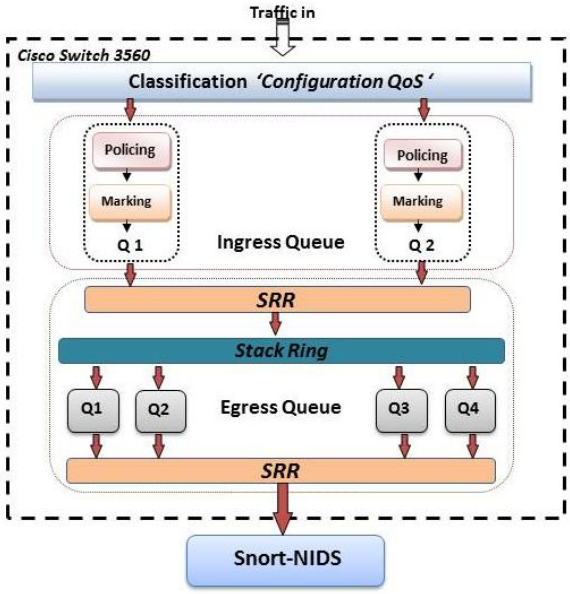

Figure 20: Snort with QoS Architecture

One of the mechanisms that QoS offers is queue technology, which can give a switch a new logical throughput-traffic-forwarding plan [26, 29]. QoS offers two input queues (ingress queues) and four output queues (egress queues) at the physical output interfaces (ports and VLANs) [26, 29].

As shown in Figure 20, we configured the switch to two input queues and four output queues, and each input queue has a policy (policy map) and marking (class map). We configured the bandwidth, threshold and priority for each input and output queue to treat traffic in the input and output queues. We also configured the speed limit for each ingress queue and egress queue using one of two functions inside the switch called Shaped or Share Round Robin (SRR) [26, 29].

The Shaped function is only available on egress queues, and a queue reserves only a portion of a port's bandwidth. SRR is available on both ingress and egress queues. It guarantees a queue a portion of a port's bandwidth, but does not limit the queue to that guaranteed amount.

However, the main idea here is to allocate a specific traffic weight and speed limit for each queue, which allows a number of packets to be sent at a specific time, thereby reducing traffic congestion even if the traffic is high-speed and heavy.

As the results shown in Figure 21 demonstrate, Snort analysed 100 percent of the traffic that reached the wire, more than 13,000kbp1ms with 0.00 percent dropped. The results show that Snort performance analyses are significantly improved when using QoS technology.

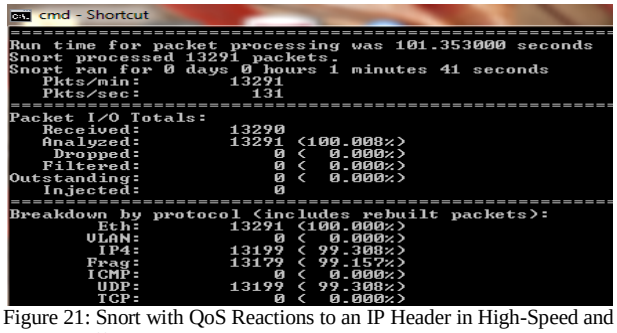

Figure 21: Snort with QoS Reactions to an IP Header in High-Speed and Heavy Traffic Networks.

E) Experiment 5. Parallel Snort-NIDS with QoS technology in high-speed network traffic

Other than the critical analysis shown in Figure 22 for experiments 1 and 4, when we sent 13,000kbp of packets at 1ms in experiment 1, Snort dropped more than 45 percent (see Figure 9); however, when we used Snort with QoS in experiment 4, Snort dropped 0.00 percent (see Figure 21). However, the difference between experiments 1 and 2 is Snort's processor times, which were 33s and 101s respectively (see Figure 22).

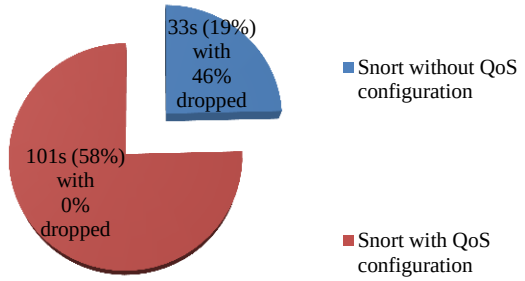

Figure 22: Snort Processor Time (13000kb/1ms)

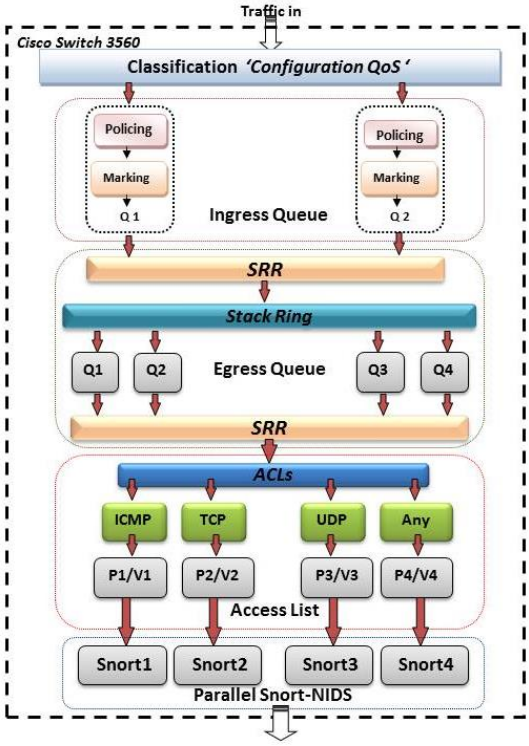

Figure 23: Architecture for Parallel Snort-NIDS Using QoS and ACLs

As a solution to reduce Snort's processor time, we suggest using parallel NIDS technology and a QoS to increase NIDS performance analysis and decrease processor time.

As we show in Figure 23, we configured and treated traffic using QoS configuration, which can parallel traffic inside output queues and then scan each queue individually using an access list function (ACL).

Using an ACL and QoS configuration, you can analyse and classify a separate type of traffic to separate the output queue. In this experiment we increased the number of packets sent to 40,000kbp/1ms.

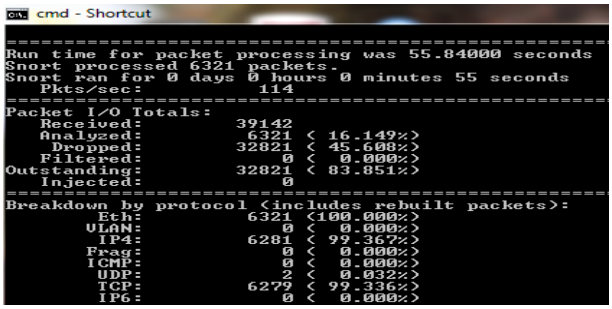

Figure 24: Snort without QoS.

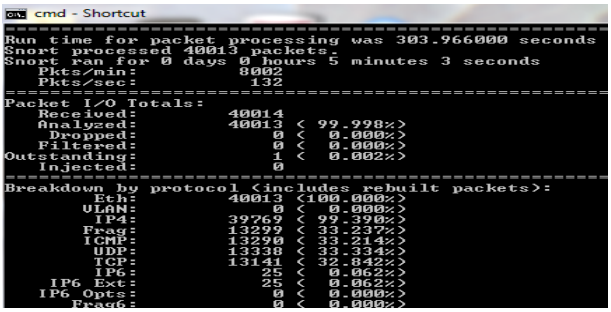

Figure 25: Snort with QoS.

As the results shown in Figures 24 and 25 demonstrate, when we tested Snort as normal without any traffic treatment, we sent nearly 40,000kb in 1 ms, Snort analysed 16 percent of the total packets received in 55s, but when we used a single Snort-NIDS with a QoS configuration and sent the same packets (40,000kb in same speed 1ms), Snort analysed all the packets that reached the wire in 302s without dropping any.

Using parallel NIDS technology (in three queues), Snort analysed 100 percent of the packets in less time (103s). Our experiments prove that Snort performance analysis improves significantly using QoS and parallel NIDs technology; it has processed more than 40,000kbp1ms in 103s with 0.00 percent dropped (see Figures 26, 27).

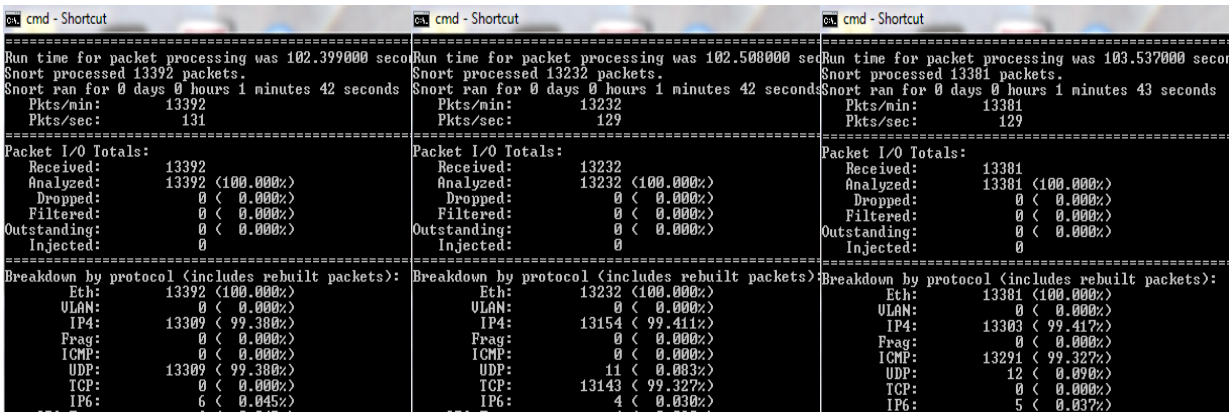

Figure 26. Paralell Snort with QoS

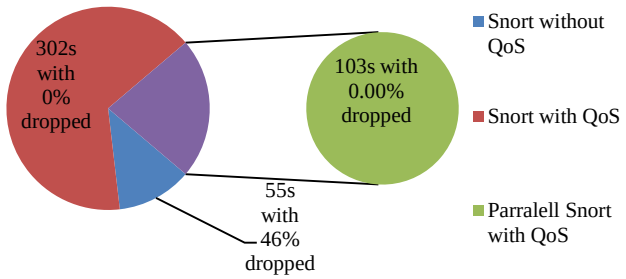

Figure 27: Snort Processor Time for 40,000kb sending in 1ms

A summary of our experiments is to increases the analytical performance of a Snort NIDS to guarantee the Quality of Service (QoS) by parallel technologies in Cisco Catalyst Switches. These experiments demonstrate the weaknesses of NIDSs, such as inability to process multiple packets and propensity to drop packets in heavy traffic and high-speed networks by incorporating various designs a real network to present experiments.

It is important to note that when you test Snort with a QoS configuration under different processors, the QoS configuration will be different depending on which type of processor is running Snort, specifically its speed (see Figure 28).

Here we tested Snort at the same speed and for the same values, but with different processors: the Intel Pentium® D CPU 2.2GHz, the Intel® corei5 2.27GHz and the Intel® corei7 2.40GHz. Snort performance analysis was affected: it performed better with the Intel® core i7 processor than the others.

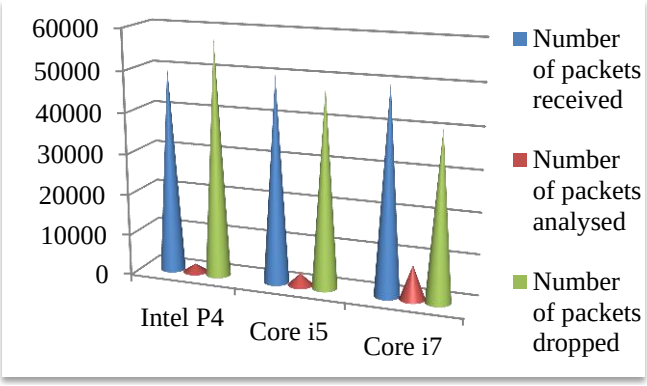

Figure 28: Snort Used with Different Processors

VIII. Related Work

Due to the fact that numerous computer systems are unable to detect or prevent threats such as DoS/DDoS attacks, the impacts of these kinds of attacks are immeasurable and irremediable. Such attackers amend, steal and destroy valuable information, and at worst damage a victim's computer system [12]; their main purpose is to stop or slow down the performance of legitimate users' computer network Systems by exploiting vulnerabilities such as mis-configuration and software bugs generated from internal and external networks [12].

Despite the existence of a variety of security protections, attackers often attempt to render services merely unavailable to intended legitimate users [16]. Here, it is insufficient to depend only on prevention techniques, especially when an attacker has successfully obtained vulnerable information from the network, but prevention can successfully and effectively restore a network before an attack is launched.

Several network-attacking techniques have been developed that enabled the researchers to conduct studies in order to prevent their network from being attacked. Many researchers are interested in IDSs due to their effectiveness in detecting attacks.

Some researchers [3, 4, 6, 11] have used distinct technologies and methodologies to detect network attacks that occur in real time, using IDSs working in high-speed network connections that have heavy traffic and open-source software like Snort or other applications to enhance the analysis and detection of data, while the others [30, 31] have used comparisons between IDS tools to achieve the best throughput results with different IDSs. Our work addresses performance in a different way. It explores the use of parallel technology to improve QoS and then NIDS performance.

Salah and Qahtan [32] implemented a hybrid scheme in Linux OS to prove that a hybrid scheme can improve the performance of general-purpose network desktops or servers running network I/O-band applications when such network hosts to both light and heavy traffic load conditions. The standard on subscribed configurations of Linux networking subsystems, as revealed by Salah and Qahtan [32], failed to meet Snort's performance level. In order to achieve a high throughput of analysed traffic with Snort, they tuned the budget parameter of the Linux Network subsystem, which controls the utilization time of the central processing unit cycle. Our work at in explores configuration at a multi-layer switch, it is in the network based rather than the host based.

Vasiliadis, Polychronakis, and Ioannidis [33] proposed a new model for a multi-parallel IDS architecture (MIDEA) for high-performance processing and stateful analysis of network traffic. Their solution offers parallelism at a subcomponent level, with NICs, CPUs and GPUs doing specialised tasks to improve scalability and running time. They showed that processing speeds can reach up to 5.2Gbit/S with zero packet loss in a multi-processor system. Their solution offers parallelism at a subcomponent level, with NICs, CPUs and GPUs doing specialised tasks. Our work has shown how QoS technology and parallelism can be impressive improving NIDS performance analyses in high speed and heavy traffic network.

These improvements in the throughput of NIDS are achieved by pairing the ASA Cisco equipment with the Snorts. Vendor companies are trying to develop security solutions to protect the enterprise network [28, 34]. Equipment, including NIDSs, has been designed to meet connectivity speed and load standards—fast network connections are more vulnerable.

Our approach differs from those of Salah and Qahtan [32], and Vasiliadis, Polychronakis, and Ioannidis [33] in that we have shown how QoS and paralleled NIDSs technology at a higher level of granularity in high-speed networks can make impressive improvements in increasing packets analysed and reducing the number of lost packets.

IX. Conclusion, Recommendations and Further Research

1. *Conclusion*

For many years attacks made on networks have risen dramatically. The major reason for this is the unlimited

access to and use of software (written and uploaded to websites by technical experts) by inadequately trained people.

Network disruptions may be caused intentionally by several types of directed attacks. These attacks are made at various layers in the TCP/IP protocol suite, including the application layer and IP. Besides the external body, attacks can be made on the network by the internal body as well.

However, an IDS is considered to be one of the best technologies to detect threats and attacks. NIDSs attracted the interest of many organizations and governments, and any Internet user can deploy them. An NIDS usually features four stages to secure a computer system network: scanning, analysing, detecting, and correcting. Our paper focused on NIDSs' weaknesses in scanning and analysing in high-speed network connectivity. We suggest using QoS configuration to improve NIDSs' analysis performance and a parallel technology to reduce NIDSs' processing time. As a result of the advance of our approach, anyone with a basic knowledge of computer networks can easily thwart an attack.

2. Recommendations

There is much yet to be learned about QoS technology. Some features of QoS may boost NIDSs performance, such as congestion management and congestion avoidance [32]. Congestion management is queuing (equal queuing), which evaluates the internal DSCP and determines which of the four egress queues in which to place the packets.

There are many items to configure when it comes to queuing: defining the priority queue, defining a queue set, guaranteeing buffer availability, limiting memory allocation, specifying buffer allocation, setting drop thresholds, mapping CoS DSCP value to queues, configuring SRR, and limiting bandwidth on an outbound queue. Congestion avoidance is used with a weighted tail drop (WTD). A lot of things in congestion avoidance may help with NIDSs performance, such as setting output queuing, configuring WTD parameters to a four-queue set, WTD thresholds for a queue, guaranteed buffer availability for a queue's maximum memory, allocation of a queue buffer for all output queues of an interface, etc.

3. Further research

This paper is centred on the failure of IDSs to prevent attacks that comes in high speed network connectively. It is having a deep consideration of experiments to present the weakness of NIDS and to improve NIDS in terms of performance, efficiency and effectiveness.

However, Multi-core technology is one solution for high-speed data and network connectively [35]. Multi-core processors provide enhancement with high capabilities and secure networks from attacks, but they increase the complexity of the security system [36, 37 and 38].

This technology is an advancement in multi-processors, which improve setup in the speed and working of the systems. Advances in the utilisation of multi-processors have yet to be exploited; however, there are two major areas of concern in computer security: the speed and volume of attacks, and the complexity of multi-stage attacks. By using multi-core processors, we can look into some of the potential technological advancements in NIDSs that can be employed for beneficial purposes and objectives.

The current and anticipated future demands for online security require the re-manipulation of existing systems and the development of improved parallel systems and in order to meet the needs of and to comply with the multi-core systems of both today and the future.

REFERENCES

[1]. A. Fuchsberger, Intrusion detection systems and intrusion prevention systems. Inf. Security Tech Rep. 10(2005) 134–139. <http://www.sciencedirect.com.lcproxy.shu.ac.uk/science/article/pii/S1363412705000415> Accessed September 14, 2013

[2]. Arbor Networks, 9th Annual Worldwide Infrastructure Security Report and ATLAS data. 2013. <http://www.arbornetworks.com/resources/infrastructure-security-report> Accessed March 25, 2014.

[3]. S. Beg, U. Naru, M. Ashraf, S. Moshin, Feasibility of intrusion detection system with high performance computing: A survey, Int. J. Advances in Computer Science 1(2010) 26–35.

[4]. M. Jamshed, J. Lee, S. Moon, I. Yun, D. Kim, S. Lee, Y. Yi, K. S. Park, Kargus, A highly-scalable software-based intrusion detection system, in: Proceedings of the 2012 ACM Conference on Computer and Communications Security, ACM, 2012, pp.317-328.

[5]. C. Wang, Z. Zhang, X. Song, Research on the information security technology of university campus network, in: D. Jin and S. Lin (Eds.), Advances in Computer Science and Information Engineering, Springer Verlag, Berlin, 2012, pp. 217–221.

[6]. F. I. Shiri, B. Shanmugam, N. B. Idris, A parallel technique for improving the performance of signature-based network intrusion detection system, in: Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference, IEEE, pp. 692–696.

[7]. M. S. Prasad, A. V. Babu, K. B. Rao, An Intrusion Detection System Architecture Based on Neural Networks and Genetic Algorithms, International Journal of Computer Science and Management Research, 2(2013) 1344-1361.

[8]. K. Scarfone, P. Mell, Guide to Intrusion Detection and Prevention Systems (IDPS): Recommendations of the National Institute of Standards and Technology, NIST Special Publication 800 (2012), 94.

[9]. V. Radhakishan, S. Selvakumar, Prevention of man-in-the-middle attacks using ID based signatures, in: Networking and Distributed Computing (ICNDC), 2011 Second International Conference, IEEE, 2011, pp. 165–169.

[10]. A. Patcha, J. Park, An overview of anomaly detection techniques: Existing solutions and latest technological trends, Computer Networks, 51(2007), 3448–3470.

[11]. W. Jiang, H. Song, Y. Dai, Real-time intrusion detection for high-speed networks, Computers and Security, 24(2005), 287–294.

[12]. W. Bul'ajoul, A. James, M. Pannu, Network intrusion detection systems in high-speed traffic in computer networks, in e-Business Engineering (ICEBE), 2013 IEEE 10th International Conference, IEEE, 2013, pp. 168–175.

[13]. Cisco Systems, Cisco Catalyst 3560 Series Switches, n.d. <http://www.cisco.com/en/US/products/hw/switches/ps5528/> Accessed October 11, 2013.

[14]. D. B. Parker, Computer Security Management, Reston Publishing Company, 1981.

[15]. G. M. Nazer, A. A. L. Selvakumar, Current intrusion detection techniques in information technology—A detailed analysis. European J. of Scientific Res., 65(2011), 611–624.

[16]. H. J. Kim, A. Pamnami, M. Patel, State-of-the-Art in Intrusion Detection Systems, Department of Electrical and Computer Engineering Stevens Institute of Technology, Hoboken, 2007.

[17]. D. Bilar, B. Daniel, Introduction to state-of-the-art intrusion detection technologies, in: Enabling Technologies for Law Enforcement, International Society for Optics and Photonics, 2001, pp. 123–133.

[18]. D. Mudzingwa, R. Agrawal, A study of methodologies used in intrusion detection and prevention system (IDPS), in: Southeastcon, 2012 Proceedings of IEEE, IEEE, 2012, pp. 1–6.

[19]. K. Scarfone, P. Mell, Guide to intrusion-detection and prevention systems (IDPS), National Institute of Standards and Technology Special Publication 800-94, February 2007. <http://csrc.nist.gov/publications/nistpubs/800-94/SP800-94.pdf> Accessed August 15, 2012.

[20]. J. Beale, B. Caswell, T. Kohlenberg, M. Poor, Snort 2.1 Intrusion Detection, second ed., Syngress Publishing, 2004.

[21]. H. Kozushko, Intrusion detection: Host-based and network-based intrusion detection systems. Independent Study. September 11, 2003.

[22]. R. Chi, Intrusion Detection System Based on Snort, In Proceedings of the 9th International Symposium on Linear Drives for Industry Applications, Springer Heidelberg, Berlin, 3(2014), pp. 657-664.

[23]. T. M. Wu, Intrusion Detection Systems, sixth ed., Intrusion Assurance Technology Analysis Center, Herndon, VA, 2009. <http://iac.dtic.mil/csiac/download/intrusion_detection.pdf> Accessed September 15, 2013.

[24]. M. Topallar, A New Host-Based Hybrid IDS Architecture—A Mind of its Own: The Know-How of Host-Based Hybrid Intrusion Detection System Architecture Using Machine Learning Algorithms with Feature Selection, VDM Verlag, Saarbrüken, Germany, 2009.
[25]. S. Staniford-Chen, C. Steven, C. Richard, D. Mark , F. Jeremy, H. James, L. Karl, W. Christopher, Y. Raymond, Z. Dan, GrIDS—a graph-based intrusion detection system for large networks, in: Proceedings of the 19th National Information Systems Security Conference, vol.1, 1996, pp. 361-370.
[26]. Cisco Systems. Understanding queuing with hierarchical queuing framework, June 2012. <http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6558/sol_ov_c22-708224.html> Accessed November 15, 2012.
[27]. M. S. Hoque, M. A. Mukit, M. A. Bikas, An implementation of intrusion detection system using genetic algorithm, Int. J. of Netw. Security & Its Applications, IV(2012), 111–112.
[28]. R. U. Rahman, Intrusion detection system with snort: Advanced IDS techniques with snort, Apache, PHP, MySQL and ACID, Pearson Education and Prentice Hall Professional, 2003.
[29]. Cisco Systems. Quality of service design overview. n.d. <http://www.cisco.com/en/US/docs/solutions/Enterprise/WAN_and_MAN/QoS_SRND/QoSIntro.html> Accessed October 20, 2013.
[30]. K.R. Karthikeyan, A. Indra, Intrusion Detection Tools and Techniques–A Survey, International Journal of Computer Theory and Engineering, 2(2010) 901-906.
[31]. P. Mehra, A brief study and comparison of Snort and Bro Open Source Netwok Intrusion Detection System, International Journal of Advanced Research in Computer and Communication Engineering, 1 (6)(2012), 383-386.
[32]. K. Salah, A. Qahtan, Implementation and experimental performance evaluation of a hybrid interrupt-handling scheme, Computer Communications, 32.1(2009), 179–188.
[33]. M. Vasiliadis, S. Polychronakis, S. Ioannidis, MIDeA: A multi-parallel intrusion detection architecture, in: Proceedings of the 18th ACM Conference on Computer and Communications Security, ACM, 2011, pp. 297–308.
[34]. Cisco Systems, Cisco ASA 5585-X Adaptive Security Appliance, n.d. <http://www.cisco.com/en/US/prod/collateral/vpndevc/ps6032/ps6094/ps6120/product_bulletin_c25-614415.html> Accessed March 12, 2013.
[35]. B. Su, Z. Xu, R. Niu, An Optimized Scheme for High-speed Data Interaction Based on TI-C6678 Multi-core DSP, In International Conference on Computer, Networks and Communication Engineering , Atlantis Press, ICCNCE, 2013, pp. 666-669.
[36]. A. Papadogiannakis, M. Polychronakis, E. P. Markatos, Scap: stream-oriented network traffic capture and analysis for high-speed networks, Proceedings of the 2013 conference on Internet measurement conference, ACM, 2013.
[37]. Y. Xiang, W. Zhou, Using multi-core processors to support network security applications, 12th IEEE Int, Workshop on Future Trends of Distributed Computing Systems, 2008 (FTDCS'08). IEEE, 2008, pp. 213-218.
[38]. T. Daxin, X. Yang, A multi-core supported intrusion detection system, IFIP Int. Conf. Network and Parallel Computing, 2008 (NPC 2008), IEEE Press, 2008, pp. 50-55.