CHAPTER 4

DETECTION OF DDoS FLOODEING ATTACKS

4.1 INTRODUCTION

The two key problems faced in the identification of malicious traffic flow are false positives and false negatives. Alerts that are triggered on normal / legitimate activity when no attack is underway is called a false positive. A false positive can be issued because the normal activity has matched an attack signature. When a detection mechanism fails to detect an actual attack, since it did not have the rules to match the attack, it is called as false negative alert.

An ideal defense system should quickly detect the onset of an attack, have the ability to correctly identify and differentiate malicious traffic from legitimate traffic, have very low or negligible false positive and false negative rates and fast response handling mechanisms. In reality this is very difficult to achieve.

4.2 D3SLR ARCHITECTURE

The proposed Dynamic DDoS Defense with Adaptive Spin Lock Rate Control mechanism consists of three functional units – Monitoring Module, Risk Assessment Module and Response Module as depicted in Figure 4.1.
Monitoring Module monitors the incoming flow at each interface of the router and passes the flow statistics to the Risk Assessment Module. Risk Assessment Module determines presence / absence of an attack flow, based on the statistics from the monitoring module and current outgoing queue length. It then generates the rate limiting rules for the malicious flow, which is implemented by the Response Module.

![Architecture of Dynamic DDoS Defense with an adaptive Spin Lock Rate control mechanism](image)

**Figure 4.1** Architecture of Dynamic DDoS Defense with an adaptive Spin Lock Rate control mechanism

### 4.2.1 Monitoring Module

Assume the defense system / router which implements D3SLR has \( N \) interfaces. Monitoring Module observes the packet arrival rate at each incoming interface for an observation interval given by \( T_{\text{obs}} \). It then calculates the collective incoming flow for each interface and computes the ratio of Collective Flow (RCF) for each interface \( (IF_i) \).

Two constant threshold values, the Maximum Queue Threshold \( (QMAX_T) \) and the Predefined Threshold value \( (PR_T) \) are also defined. The Maximum Queue Threshold is set to 75 percentage of the length of the router’s outgoing queue.
To detect and mitigate a DDoS attack in an intermediate node in the transit network, it is essential to determine the threshold for that network, which defines the volume of traffic that can be effectively handled by the intermediate node. The predefined threshold value is the volume of acceptable traffic per protocol at each interface such that the total volume of traffic from all interfaces does not exceed the load on the outgoing queue. This depends on the router capacity and number of incoming interface and type of protocol for which the threshold value is to be calculated. In the testbed simulation the threshold is set to four times the legitimate traffic rate. The traces from USC / ISI ANT Datasets have traffic information for a short duration prior to and after the attack and twice this value is taken as the threshold.

When measurement activities are initiated by the Risk Assessment Module, the Monitoring Module monitors the incoming packets and updates a three dimensional data structure involving a Destination Based List (DBL), Source Based List (SBL) and Protocol Based List (PBL).

All Lists use a Time Stamp (TS) field to monitor when a record was last modified. When a memory constraint occurs, the record with the oldest TS (Least Recently Used) is replaced.

Destination Based List is indexed by the Destination Address (DA) of all packets arriving at infected interface. Source Based List is indexed by the Source Address of the machines generating the packets for the Destination Address from the DBL. For a specific Source Address – Destination Address (SA-DA) pair the Protocol Based List records the count of various protocol type of traffic.
4.2.2 Risk Assessment Module

The goal of the Risk Assessment Module is to identify low level observation like significant rise in load at router, packet volume, etc., into high level normal / suspicious / attack activity assertion without relying on probability parameters. Each observation is mapped to a logical conclusion. For example a sudden abnormal spike in traffic volume may indicate a suspicious activity, whereas, a constant high volume of traffic towards a machine indicates malicious activity targeted at machine.

At the end of every observation interval $T_{obs}$ Risk Assessment Module receives the flow statistics information from the Monitoring Module and the current load on outgoing queue (Q_CUR) from the Response Module. Based on this information the Risk Assessment Module classifies the type of traffic flow as Legitimate, Suspicious or Malicious.

If the ratio of Collective Flow for an interface is less than a predefined threshold value and load on the outgoing queue is below the maximum queue threshold, Risk Assessment Module classifies the incoming flow as Legitimate Flow and the interface is tagged as Safe.

If the ratio of Collective Flow for an interface is above the predefined threshold value and load at the outgoing queue is less than the maximum queue threshold, flow is classified as Suspicious Flow and the interface is tagged as Suspect.

If the ratio of Collective Flow for an interface is above the predefined threshold value and load at the outgoing queue exceeds the maximum queue threshold, flow is confirmed as Malicious Flow and the interface is tagged as Infected.
When a flow is classified as suspicious flow the Risk Assessment Module activates individual packet monitoring and measurement activities at the suspected interface by the Monitoring Module. If the flow is later confirmed as malicious, the Risk Assessment Module defines the rate limiting rules and initiates rate limiting activity on the malicious flow at the infected edge.

4.2.3 Response Module

Response Module monitors the load on the outgoing queue and periodically passes the information to the Risk Assessment Module. It receives the alerts generated by the Risk Assessment Module, correlates and prioritizes the alerts and then executes the rate limiting rules.

4.3 PACKET OBSERVATION AND MEASUREMENTS

Packet monitoring in this thesis is performed using a Divide and Conquer approach and an iterative refinement is applied in successive observation intervals to identify the malicious activity.

1. All incoming packets at router are not monitored. Rather the incoming flow at suspected / infected interface alone is monitored.

2. Rate limiting is implemented only on confirmation of malicious flow.

3. Iterative refinement is used to identify the target of DDoS attack, the source machine generating the malicious traffic and packet type of the malicious traffic and alerts are generated.
Based on this information, rate limiting rules are generated and implemented on the malicious traffic from the infected interface, while legitimate traffic from the infected interface is left relatively undisturbed.

4.3.1 Packet Observation

Internally packet observation entails a sniffer, preprocessor and alert subsystems. The packet sniffer component reads the packet profile of each packet received at the interface. Packet profiles include the source and destination address, ports used, protocol used, packets per second, time interval between packets sent, etc. The Lists which store the packet profiles are centrally administrated and dynamically updated. The sniffer then decodes the packet to a format that can be understood and used by the preprocessor. The preprocessor detects the presence or absence of a malicious flood and generates the flood alerts. The alert subsystem manages the security log that logs the alerts generated by the preprocessor.

4.3.2 Traffic Measurement

D3SLR is configured to monitor the incoming traffic at each of its interfaces. When the Risk Assessment Module identifies a suspicious flow, it initiates a measurement activity at the Monitoring Module for the suspected interface. The Destination Based List (DBL) update is first initiated and updated to determine

1. The incoming interface at which the packet were received,
2. The destination address of the packets received,
3. Number of packets received per destination address in that observation interval and
4. The timestamp of last update for each record of the List.
The **Source Based List (SBL)** is updated for each specific destination address identified from the Destination Based List. SBL is indexed by the source address of the machines generating the malicious traffic towards the potential victim at the infected interface. Per observation interval, SBL records

1. The IP address of the source machines generating the packets,
2. The IP address of the destination machine,
3. The number of packets received for the destination address from the source machines and
4. The timestamp of last update for each record of the List.

The types of protocol used by the source machine from Source Based List for generating the malicious traffic towards the target machine identified by the Destination Based List is determined by the **Protocol Based List (PBL)** which records the count of various protocol type of traffic. Per observation interval the Protocol Based List monitors the

1. The IP address of the source machines generating the packets,
2. The IP address of the destination machine,
3. Protocol of the packet
4. The number of packets sent and received for the destination address from the source machines per protocol type
5. The number of bytes sent and received for the destination address from the source machines per protocol type and
6. The timestamp of the last update for each record of the List.

An interface is classified as Safe, Suspicious or Infected based on the volume of malicious traffic received at that interface. At the end of the
observation interval the potential target machine is identified by the abnormal volume of the malicious traffic received towards that machine at the infected interface.

The Source Based List isolates the source machine(s) generating the malicious traffic towards the victim. The Protocol Based List identifies the protocol used by the source machine(s) to generate the malicious traffic towards the victim machine and defines an attack signature consisting of Source Address, Destination Address and Protocol-type of the packet.

At the end of each observation interval, all entries in the Lists are cleared and the values are recalibrated.

4.4 FLOOD ATTACK DETECTION

Traffic flow may be characterized as unidirectional (UDP / streaming media) or bidirectional (TCP, ICMP_ECHO and DNS). Unidirectional flows will send traffic to a single destination without requiring any reply while bidirectional flows typically exhibit request / reply dynamics.

UDP flows transmit data to a destination without requiring any replies or acknowledgement. TCP flows, on the other hand, involve data packets flow in one direction from sender to receiver with ACKs in the opposite direction from receiver to sender. Requests to the DNS server and ICMP_ECHO request to any host will also invoke a reply in the reverse direction.

The DDoS flood attack detection in D3SLR relies on presence of persistent aggressive traffic flow towards a single destination. Attack detection is implemented as a Snort plugin module, called the Flood Detection Preprocessor.
4.5 IMPLEMENTATION OF DETECTION COMPONENT

In order to deploy Snort to detect flooding attacks, a Snort plugin module, the Flood Detection Preprocessor is created and additionally introduced in the Snort code directory. The Flood Detection Preprocessor monitors the incoming traffic and quickly detects the onset of a malicious flood attack.

4.5.1 Flood Detection Preprocessor

The Snort Flood Detection Preprocessor plugin module includes a header file and a source file placed in the $SNORT_DIR/src/preprocessor. The preprocessor header file contains variables which are to be initialized to default values, including the default network threshold, observation interval and logfile, data structures to be used in the plugin module and the function prototypes. Some of the key data structures used in D3SLR are shown in Figure 4.2.

The Flood Detection Preprocessor maintains the packet count using a three dimensional double linked list: DSPList containing a DBL (match by Destination IP address), SBL (match by source IP address) and PBL (match by protocol type of the packet). For a given incoming interface, the destination address for which the packets are transmitted, the packet’s source address for that destination address, the protocol type for the above source – destination address pair and the total packet count are registered.
Figure 4.2 User Defined Data Structures
The function prototypes include the initialization and core functions of the source file and standard functions for exit and restart which are to be shared with other preprocessors also. The source file specifies the header files to be included and also defines the internal functions specifying what to do when a packet is received. This forms the core of the user defined preprocessor plugin. Some of the core user defined functions are:

- voidRecvPkt(Packet *p);

  When a packet is received count the total number of packets received for each interface. At the end of the observation interval, calculate the ratio of Collective Flow (RCF) and check the load on the outgoing queue. Classify the interface as Safe, Suspicious or Infected.

- voidupdateList(DSPList *dsplist);

  If an interface is tagged as suspicious or infected, initiate measurement activity of individual packets and update the data structures. For a given packet, parse the destination address, source address and protocol type of the packet. Traverse the data structure to check if an entry already exists. If an entry already exists increment the packet count and update the time stamp (TS) of the record.

  If no entry exists, create and insert a new record in the data structure. If there is a memory constraint, set the memErr_Flag and call the ClearList() function to clear the least recent used record to create memory space for the new record.
- void clearList(DSPList *dsplist);

The clearList() function is automatically invoked

  i. At the end of each observation interval – delete all entries and reset the data structure.

  ii. When there is a memory space constraint and the memErr_Flag is set – remove the record with the least time stamp (TS), to allow new entries to be made into the data structure.

- intchkFlood(DSPList *dsplist, Packet *p);

The chkFlood() function checks the packet count for a given protocol type between a Destination – Source pair.

### 4.5.2 Detecting Non–Responsive Source Machines

An Aggressive one-way traffic on a protocol or application that is expected to exhibit bi–directional communication pattern is typically regarded as a sign of an attack. Consider for example, a TCP packet transmission from a sender with corresponding reply and / or acknowledgements from the receiver.

If the sender is a legitimate source, the ratio of packet sent to acknowledgement received from the receiver will typically be close to 1. In case the receiver is under attack and is unable to send the acknowledgements, a legitimate sender will assume that congestion has occurred in the network and adapt the packet transmission rate to the network state. Hence the ratio of packet sent to acknowledgement received will still remain closer to 1.
A malicious sender on the other hand transmits a high rate of packet and this rate persists even in the absence of acknowledgement. The ratio of packets sent to acknowledgements received from the receiver will now be higher than 1 and if the transmission rate persists for a long enough duration, the ratio will exceed a threshold and flow is classified as an attack.

a. TCP and ICMP Flood Attacks

This thesis detects aggressive TCP and ICMP Flood attacks by determining the number of TCP and ICMP packets sent and received on a connection and calculating the ratio as a weighted average over a period of time. If the increase in the ratio is consistent, it will exceed the given threshold and the flow is classified as malicious and all its packets are classified as attack and a flood alert is generated.

b. TCP SYN Attacks

The SYN attack is a special case of bi-directional attack, which consumes the victim’s connection buffer by sending too many TCP SYN packets that invoke the victim to allocate a new record in the connection buffer for each SYN. The attacker never completes the three-way handshake and the victim’s state will eventually time out leading to deletion of stale records, but the timeout period is very long. A moderate-rate attack can thus keep victim’s resources effectively bound and deny service to new connections on the attacked port.

This thesis detects TCP SYN attacks by keeping a record of SYN-to-ACK ratio for each destination of TCP traffic. When this ratio crosses the threshold value, a flood attack alert is signaled at the destination machine.
4.5.3 Detecting UDP Flooding Attacks

Numerous key Internet applications use UDP, including the Domain Name System (DNS), where queries must be fast and only consist of a single request followed by a single reply packet, the Simple Network Management Protocol (SNMP), the Routing Information Protocol (RIP) and the Dynamic Host Configuration Protocol (DHCP).

Voice and video traffic is generally transmitted using UDP. Real-time audio and video streaming protocols are designed to handle occasional lost packets, so only slight degradation in quality occurs, rather than large delays if lost packets were retransmitted. The transmission rate of UDP packets hence depends on the underlying application and varies from low rate of request / reply queries to high rate of audio and streaming data.

DNS flooding attack is similar to bi-directional TCP and ICMP flood attack and handled likewise in this thesis, by calculating the weighted average of the ratio of packet sent and received over a period of time. If the increase in the ratio is consistent, it will exceed the given threshold and the flow is classified as malicious and all its packets are classified as attack and flood alert is generated. In case of a flooding attack using streaming media, the volume of streaming data will be severe enough to cause congestion.

This thesis applies a rate limiting on such streaming media, by defining the allowed number of connections per destination and the number of packets per connection.

4.6 PERFORMANCE EVALUATION

To evaluate the attack detection component of the proposed Dynamic DDoS Defense with an adaptive Spin Lock Rate control
mechanism, a DDoS attack topology was created and the detection component was evaluated in a number of different situations involving various combinations of protocols used for generating the legitimate and attack traffic. The various metrics under which the component was evaluated includes duration of attack, volume of legitimate and malicious traffic generated, attack rate and number of legitimate and attack hosts.

The scale of a DDoS network is too large to be recreated in a laboratory environment. So a testbed was setup to simulate a DDoS topology and attacks were deployed with varying traffic characteristics to determine the effectiveness of the defense system in detecting an attack and identifying the legitimate and malicious traffic.

4.7 EVALUATION GOALS

A variety of synthetic attacks involving UDP flood, TCP SYN flood and ICMP flood attacks were executed in the testbed and the generated traces were recorded. Data traces created from such controlled setup in testbed environments do not depict the actual scenario. The goal of developing such models is to apply D3SLR to the data traces and evaluate its accuracy in detecting attacks and separating the legitimate from the attack traffic and test whether the model developed can be generalized to identify attack traces in various datasets obtained from different techniques.

4.8 TESTBED EVALUATION

The three important aspects of a testbed deployment are the test topology, network traffic (both legitimate and malicious) and the attack characteristics.
4.8.1 Test Topology

The testbed topology to simulate a DDoS attack scenario consists of a Handler machine, three Agent machines, one legitimate host and a target machine. The Agents are connected to victim / target machine via switches. The setup was isolated to ensure that the testbed traffic was not leaked into the Internet during testing.

The testbed topology is as illustrated in Figure 4.3. Attacks involving various combinations of UDP flood, TCP SYN flood and ICMP flood attacks were executed in the testbed and the traces were recorded from the four Agent machines. The IP addresses of the machines were varied to simulate individual host flooding, in which all four Agent machines were assigned IP address with different network address and subnet flooding, in which all four Agent machines were assigned the same network address.

D3SLR was applied to these traces to evaluate the accuracy of D3SLR in detecting attacks and separating the legitimate traffic from the attack traffic.

Figure 4.3 Test Bed Topology
4.8.2 Network Traffic

Attack dynamics include packets with constant or variable packet size and the rate of packets generated can be constant or can be gradually increased or decreased between two limits. To simulate DDoS attack UDP flood, TCP SYN flood and ICMP flood were generated.

Attack traffic between the Agents and target machine was generated using Stacheldraht. A small volume of legitimate traffic was also generated simultaneously for each type of protocol from various IP address.

4.8.3 Attack Characteristics

Several groups of test scenarios involving various combinations of UDP, TCP SYN and ICMP protocol for generating the attack traffic, were conducted with the testbed. The duration of attack and packet transmission rate were also varied for each test run and results were compiled. For each scenario, a minimum of three test runs was executed to verify performance consistency. Since the results were relatively consistent, only the last set of performance results are presented in this section.

4.9 EVALUATION RESULT

To simulate DDoS attack in the testbed, UDP flood, TCP SYN flood and ICMP flood were generated using Stacheldraht tool. The steps to generate the mixture of TCP SYN, ICMP_ECHO and UDP attack packets using Stacheldraht tool are detailed in Appendix 2.

Size of the attack packets, the rate of packets generated and duration of attacks were varied. The effect of the attack was similar regardless of the protocol used. The data collected using Libpcap at the defense system is presented in Table 4.1.
### Table 4.1 Testbed Environment

<table>
<thead>
<tr>
<th>PARAMETERS</th>
<th>ATTACK DYNAMICS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attack</td>
<td>Scenario – 1</td>
</tr>
<tr>
<td>Duration of simulation</td>
<td>300 seconds</td>
</tr>
<tr>
<td>Observation Interval</td>
<td>1 second</td>
</tr>
<tr>
<td>Protocol used</td>
<td>UDP</td>
</tr>
<tr>
<td>Attack rate</td>
<td>Constant</td>
</tr>
<tr>
<td>Number of legitimate hosts</td>
<td>1</td>
</tr>
<tr>
<td>Number of attack hosts</td>
<td>3 hosts</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PARAMETERS</th>
<th>STATISTICS OF PACKETS RECEIVED AT DEFENSE SYSTEM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Packets</td>
<td>95718</td>
</tr>
<tr>
<td>Number of Bytes</td>
<td>5821983</td>
</tr>
<tr>
<td>Average packet size</td>
<td>60 bytes</td>
</tr>
<tr>
<td>Average Packet/second</td>
<td>319</td>
</tr>
<tr>
<td>Average Bytes/second</td>
<td>19407</td>
</tr>
</tbody>
</table>
In the first scenario, three host machines were used to generate UDP attack traffic. All three machines generated packets of size 60 bytes, at approximately 300 packets / second towards the victim machine. The source and destination port numbers were randomized during the test run. The observation interval was set at 1 second. The source address of attacking hosts and the attributes of the attack traffic are shown in Table 4.2.

### Table 4.2 Attack Traffic Characteristics : Scenario – 1

<table>
<thead>
<tr>
<th>Source IP</th>
<th>Attack Type</th>
<th>Duration (Seconds)</th>
<th>Number of Packets</th>
<th>Packet Rate (Packets/Second)</th>
<th>Byte Rate (Bytes/Second)</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.95.27.190</td>
<td>UDP Flood</td>
<td>300</td>
<td>25935</td>
<td>87</td>
<td>5129</td>
</tr>
<tr>
<td>192.120.148.227</td>
<td>UDP Flood</td>
<td>300</td>
<td>19323</td>
<td>64</td>
<td>3893</td>
</tr>
<tr>
<td>202.1.175.252</td>
<td>UDP Flood</td>
<td>300</td>
<td>18840</td>
<td>63</td>
<td>3796</td>
</tr>
</tbody>
</table>

The attack started at 0.004560 seconds and stopped at 299.993309 seconds and was detected at 1.009226 seconds with a detection delay of 1.004666 seconds.

In the second scenario, three Agent machines (IP : 133.172.248.56, 132.244.39.212 and 126.120.0.39) were used to generate TCP SYN requests with constant sequence number at approximately 225 packets / second towards the victim machine. The same source port and destination port numbers were used throughout the trace. The observation interval was set at 2 seconds. The source address of attacking hosts and the attributes of the attack traffic are shown in Table 4.3.
Table 4.3 Attack Traffic Characteristics: Scenario – 2

<table>
<thead>
<tr>
<th>Source IP</th>
<th>Attack Type</th>
<th>Duration(Seconds)</th>
<th>Number of Packets</th>
<th>Packet Rate (Packets/Second)</th>
<th>Byte Rate (Bytes/Second)</th>
</tr>
</thead>
<tbody>
<tr>
<td>133.172.248.56</td>
<td>TCP SYN Flood</td>
<td>121</td>
<td>9813</td>
<td>81</td>
<td>36199</td>
</tr>
<tr>
<td>132.244.39.212</td>
<td>TCP SYN Flood</td>
<td>133.8</td>
<td>2776</td>
<td>21</td>
<td>6466</td>
</tr>
<tr>
<td>126.120.0.39</td>
<td>TCP SYN Flood</td>
<td>153</td>
<td>24731</td>
<td>161</td>
<td>49455</td>
</tr>
</tbody>
</table>

Attack traffic generation from machine 133.172.248.56, 132.244.39.212 and 126.120.0.39 started at 59.793319, 58.001868 and 64.494470 seconds and lasted for 121, 134 and 153 seconds respectively from the three machines. The attacks from machine 1 and 2 were detected at 60 seconds and from machine 3 at 6 seconds. The average detection delay for the three attacks was 1.236781 seconds.

In the third scenario, three Agent machines were used to generate UDP, TCP SYN and ICMP flood towards the victim machine. The same source port and destination port numbers were used throughout the trace. The source address of attacking hosts and the attributes of the attack traffic are shown in Table 4.4.
Table 4.4 Attack Traffic Characteristics : Scenario – 3

<table>
<thead>
<tr>
<th>Source IP</th>
<th>Attack Type</th>
<th>Duration (Seconds)</th>
<th>Number of Packets</th>
<th>Packet Rate (Packets/Second)</th>
<th>Byte Rate (Bytes/Second)</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.120.148.227</td>
<td>UDP Flood</td>
<td>600</td>
<td>41588</td>
<td>69</td>
<td>4187</td>
</tr>
<tr>
<td>192.95.27.190</td>
<td>TCP SYN Flood</td>
<td>600</td>
<td>51564</td>
<td>86</td>
<td>5099</td>
</tr>
<tr>
<td>40.75.89.172</td>
<td>ICMP Flood</td>
<td>600</td>
<td>36038</td>
<td>60</td>
<td>3632</td>
</tr>
</tbody>
</table>

Attack traffic generation at all three machines continued for 600 seconds. The rate of packet generation was kept constant at all three machines. The observation interval was set at 3 seconds. The attacks were detected at 3.001657 with a detection delay of 1.038521 seconds.

In the fourth scenario, three Agent machines were used to generate UDP, TCP SYN and ICMP flood towards the victim machine. The same source port and destination port numbers were used throughout the trace. The source address of attacking hosts and the attributes of the attack traffic are shown in Table 4.5.
Table 4.5 Attack Traffic Characteristics: Scenario – 4

<table>
<thead>
<tr>
<th>Source IP</th>
<th>Attack Type</th>
<th>Duration (Seconds)</th>
<th>Number of Packets</th>
<th>Packet Rate (Packets/Second)</th>
<th>Byte Rate (Bytes/Second)</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.95.27.190</td>
<td>UDP Flood</td>
<td>1200</td>
<td>100366</td>
<td>84</td>
<td>4962</td>
</tr>
<tr>
<td>51.173.229.255</td>
<td>TCP SYN Flood</td>
<td>1200</td>
<td>77635</td>
<td>65</td>
<td>3910</td>
</tr>
<tr>
<td>40.75.89.172</td>
<td>ICMP Flood</td>
<td>1200</td>
<td>71964</td>
<td>60</td>
<td>3627</td>
</tr>
</tbody>
</table>

The attack traffic generation at all three machines started at 0 second and ended at 1200 seconds. The rate of packet generation was increased by 0.5% every 50 seconds. The observation interval was set at 5 seconds. The attacks were detected successfully at 5.002317 seconds with a detection delay of 2.003751 seconds.

The volume of malicious traffic generated in the first testbed scenario for UDP attack was thrice the volume of legitimate traffic. TCP SYN attack requires lesser number of packets since it targets the victim’s connection resources rather than its bandwidth and hence for the second testbed scenario involving TCP SYN attack, volume of malicious traffic was set only to twice the legitimate traffic. For the third and fourth test bed scenarios, the duration of attack was extended and malicious traffic volume was set to twice the legitimate traffic.
It was observed that the observation interval was a key factor in the detection delay. The longer the duration of observation interval, the detection was more accurate, but the detection delay was higher. A shorter observation interval resulted in smaller detection delay but created more overhead in alert generation.

4.10 CONCLUSION

The success of the DDoS attack against a defense system and in turn the victim is defined by the volume of false negatives and false positives at the defense system. The proposed defense system can be deployed at critical points in the network to detect the onset of the DDoS flood attacks quickly.

By initiating individual packet measurement and log activity only on confirmed detection of DDoS activity, D3SLR successfully eliminates the computational and memory overhead issues of the DWARD system.

At the end of every observation interval, an alert is generated if a flooding attack is detected by the flood detection preprocessor. Each alert has many attributes including the destination address, destination port, source address, source port, protocol type of the packet, time stamp of alert generation, priority level and the rate limit to be applied.

Since a DDoS attack involves many source machines generating the malicious packets towards one target machine, the number of alerts generated may be high. Also if the attack duration is longer than one observation interval, the alerts are repeated in each observation interval. To address these issues, the alerts are analyzed, merged and prioritized based on the duration of the attack and the severity of the malicious flow and an
adaptive rate limiting is applied to the traffic flow from the identified source to destination.

The following chapter discusses the alert correlation and prioritization issues to organize the alerts for effective rate limiting of malicious traffic.