CHAPTER 4
A DISTRIBUTED APPROACH TO DEFEND WEB SERVICE AGAINST DDOS ATTACKS

Most of the business applications on the Internet are dependent on web services for their transactions. Distributed denial of service (DDoS) attacks either degrade or completely disrupt web services by sending flood of packets and legitimate looking requests towards the victim web servers. The detection of DDoS attacks is one of the hardest problems confronted by the network security researchers. Flash Event (FE), which is caused by a large number of legitimate requests, has similar characteristics as that of DDoS attacks. Moreover DDoS attacks and FEs require altogether different handling procedures. So discriminating DDoS attacks from FEs is very important. But the research involving DDoS detection has not laid enough emphasis on including FEs scenarios in the experiments. An array of defense schemes are proposed but still defending web service from DDoS attacks as well as discriminating it from FE is largely an unsolvable problem so far. In this chapter, traffic cluster entropy is conceptualized from source address entropy and their combination is not only used to detect various types of DDoS attacks against the web service, but also to distinguish DDoS attack from FEs. Various modules of DDoS defense system such as detection, characterisation and filtration of attack traffic are proposed and their deployment on the Internet is also discussed. Finally an integrated, dynamic and autonomous framework is proposed which can detect the attack, characterize attack sources, and filter the attack packets as early as possible so as to minimize the collateral damage.

4.1 Traffic Feature Distributions and Shannon Entropy

Networks today increasingly see unusual traffic patterns. These traffic patterns arise from network abuse such as DDoS, port scans, and worms as well as from legitimate activity such as transient changes in customer demand, flash crowds, or occasional high-volume flows. The events caused due to network abuse are known as network based anomalies. A range of these network anomalies have been witnessed on the Internet routinely. Some of these network-based anomalies are
malicious and become major threats to network security. These threats have led to a steady need for development of countermeasures. The countermeasures normally include monitoring of traffic followed by sophisticated analysis so that abnormal events can be pinpointed. In reality, network traffic contains a wealth of information about normal and abnormal traffic behaviour. The recognition of anomalies in the time domain is difficult because they are buried within the other traffic. An important challenge therefore is to determine how to extract understanding about the presence and nature of traffic anomalies from the potentially overwhelming mass of network-wide traffic data.

Lakhina et al. (2004) have highlighted that a lot of work has been done in which volume (no. of bytes, packets and flows) is chosen as a principal metric for detecting anomalies. Volume based detection schemes can detect anomalies that causes large traffic changes. But anomalies like low rate DDoS attacks which do not cause much change in traffic volume cannot be detected accurately. These low rate DDoS attacks and other network anomalies however can be detected by analysing distribution of traffic features. A traffic feature is a field in the header of the packet. A metric is a descriptive statistic that can be calculated from collected traffic features of one or more packets in traffic. The examples of metric can be mean packet length, mean rate of incoming and outgoing traffic, distribution of IP addresses etc. Some of the well known traffic events and traffic feature distribution affected by these events are listed below in table 4.1.

It is clear from the table that traffic feature distributions are affected due to network anomalies. So, these distributions in general are helpful in detecting various kinds of network anomalies. However the focus of our work is on detecting DDoS attacks and distinguishing the same from flash events. DDoS attacks are network-wide attacks. The special feature of current DDoS attack packets is that individually each packet is a perfect legitimate packet but in combination, correlating these packets monitored at different points can give some signs of uniqueness from legitimate packets. In order to detect DDoS attacks packets headers, packets aggregate flows, and/or correlations are analysed with the aim of distinguishing normal traffic from the attack traffic. As per Yuan and Mill (2005) and
<table>
<thead>
<tr>
<th>S. No.</th>
<th>Event Label</th>
<th>Nature of event (Legitimate/Anomaly)</th>
<th>Definition</th>
<th>Traffic Distributions Affected (skewed/dispersed)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Alpha Flows</td>
<td>Legitimate</td>
<td>Unusually large volume point to point flow.</td>
<td>Skewed traffic distribution of source and destination addresses (possibly ports).</td>
</tr>
<tr>
<td>2.</td>
<td>Flash Crowd</td>
<td>Legitimate</td>
<td>Unusual burst of traffic to single destination, from a “typical” distribution of sources.</td>
<td>Skewed traffic distribution of destination addresses and destination ports. Dispersed traffic distribution of source addresses.</td>
</tr>
<tr>
<td>3.</td>
<td>Port Scan</td>
<td>Anomaly</td>
<td>Probes many destination ports on a small set of destination addresses.</td>
<td>Skewed traffic distribution of destination addresses and dispersed traffic distribution of destination ports.</td>
</tr>
<tr>
<td>4.</td>
<td>Network Scan</td>
<td>Anomaly</td>
<td>Probes many destination addresses on a small set of destination ports.</td>
<td>Skewed traffic distribution of destination ports and dispersed traffic distribution of destination addresses.</td>
</tr>
<tr>
<td>5.</td>
<td>Outage Events</td>
<td>Legitimate (Network Fault)</td>
<td>Traffic shifts due to equipment failures or maintenance.</td>
<td>Depending upon types of unintentional failures in the network, different traffic distributions are affected.</td>
</tr>
<tr>
<td>6.</td>
<td>Point to Multipoint</td>
<td>Legitimate</td>
<td>Traffic from single source to many destinations, e.g., content distribution.</td>
<td>Skewed traffic distribution of Source address and dispersed traffic distribution of destination addresses.</td>
</tr>
<tr>
<td>7.</td>
<td>Worms</td>
<td>Anomaly</td>
<td>Scanning by worms for vulnerable hosts (special case of Network Scan)</td>
<td>Skewed traffic distribution of destination ports.</td>
</tr>
<tr>
<td>8.</td>
<td>DOS/DDoS</td>
<td>Anomaly</td>
<td>An intentional attempt to deny service by exploiting the bugs in the victim machine or exhausting victim’s resources through sheer flooding of traffic from lot of sources.</td>
<td>Skewed traffic distribution of destination addresses and dispersed traffic distribution of source addresses.</td>
</tr>
</tbody>
</table>
Lawniczak et al. (2009), packet traffic is often analysed near the victim or near the attack sources. As DDoS attacks are purposely created by humans, they must affect the natural “randomness” and “natural structure and order” of packet traffic under normal conditions (Nucci and Banneman, 2007). This randomness of network traffic can be measured with the help of a statistic called Shannon entropy. In information theory, Shannon entropy or information entropy is a measure of the uncertainty associated with a random variable (Shannon, 2001). This entropy can be used to calculate the distribution randomness of some attributes in the network packets’ headers. These attributes could be the packet’s source IP address, TTL value, or other packet header fields. Even management information base (MIB) variables and resource usage variables can also be used for the purpose. For example, the detector captures 1000 continuous data packets at a peak point, and calculates the frequency of each distinct IP address among these 1000 packets. By further calculation of this distribution, we can measure the randomness of these packets (Feinstein and Schnackenberg, 2003).

In this work, source IP address is used as a traffic feature to detect DDoS attacks. Attackers always recruit a lot of zombies all around the Internet to launch DDoS attack. As per Peng et al., 2007, the number of zombies in a typical DDoS attack can vary from one hundred to more than 100,000. These zombies either use own or spoofed IP addresses. The use of large number of zombies and spoofed addresses trigger change in source IP distribution. Thus, with analysis of source address distribution, DDoS attacks can be detected easily (Feinstein et al., 2003; Kumar et al., 2007; Li et al., 2009; Sardana, 2009). However, source address distribution is not sufficient to distinguish DDoS attacks from flash events as in most of the situations DDoS attacks and flash events affect source IP distributions in similar ways.

As per table 4.1, two properties of traffic distributions namely dispersion and skewness are used to characterize attack anomalies. A metric that captures the degree of dispersal or concentration of a distribution is called sample entropy (Shannon and Weaver, 1963).
Let \( X = \{n_1, i = 1, \ldots, N\} \) be the frequency distribution consisting of \( N \) features
where feature \( i \) occurs \( n_i \) times in the sample.

Let \( S = \sum_{i=1}^{N} n_i \) be the total number of observations in the distribution. It actually
represents total number of packets observed in the sample.

Let \( p_i = n_i / S \) be the probability of occurrence each feature \( i \) in the sample.

Then, the sample entropy \( H(X) \) is calculated as

\[
H(X) = - \sum_{i=1}^{N} (p_i) \times \log_2(p_i) \quad (4.1)
\]

Where \( X \) is the frequency distribution consisting of \( N \) features. It actually
gives number of packets observed \( n_i \) for each feature \( i \) in the sample.

The value of sample entropy lies in the range \( (0 - \log_2 N) \). The metric takes
on the value 0 when the distribution is maximally concentrated, i.e. all observations
are the same. Sample entropy takes on the value \( \log_2 N \) when the distribution is
maximally dispersed, i.e. \( n_1 = n_2 = \ldots = n_N \). Figure 4.1 depicts maximum sample
entropy values for number of flows. For example, consider a hypothetical network in
which each source sends an average of 5 packets under normal conditions. With 16
such different sources, total number of packets is 80, according to equation 4.1, the
system entropy \( H(X) \) under normal condition is as below:

\[
\begin{align*}
H(X) &= -\sum_{i=1}^{16} \left( \frac{5}{80} \right) \log_2 \left( \frac{5}{80} \right) \\
H(X) &= -\sum_{i=1}^{16} \left( \frac{1}{16} \right) \log_2 \left( \frac{1}{16} \right) \\
H(X) &= -\sum_{i=1}^{16} \left( \frac{1}{16} \right) \log_2 (2)^{-4} \\
H(X) &= -\sum_{i=1}^{16} \left( \frac{4}{16} \right) \\
H(X) &= -16 \times \left( \frac{4}{16} \right) \\
H(X) &= 4
\end{align*}
\]
Consider a concentrated attack where there is a single source sending 80 packets and all other 15 sources send no packets at all. The system entropy according to equation 4.1 is as below:

\[ H(X) = -(\frac{80}{80}) \log_2(\frac{80}{80}) - \sum_{i=1}^{15} \left( \frac{0}{80} \right) \log_2 \left( \frac{0}{80} \right) \]

\[ H(X) = -(1) \log_2 (1) - 0 \]

\[ H(X) = -(1) \log_2 (2)^0 \]

\[ H(X) = 0 \]

Next consider a distributed attack where there are 32 sources sending 2 packets each along with normal traffic from 16 sources sending 4 packets each. The total number of sources is 48 and total number of packets is 128. The system entropy as per equation 4.1 is given below: -

\[ H(X) = -\sum_{i=1}^{32} \left( \frac{2}{128} \right) \log_2 \left( \frac{2}{128} \right) - \sum_{i=1}^{48} \left( \frac{4}{128} \right) \log_2 \left( \frac{4}{128} \right) \]

\[ H(X) = -\sum_{i=1}^{32} \left( \frac{1}{64} \right) \log_2 \left( \frac{1}{64} \right) - \sum_{i=1}^{48} \left( \frac{1}{32} \right) \log_2 \left( \frac{1}{32} \right) \]

\[ H(X) = -\sum_{i=1}^{32} \left( \frac{1}{64} \right) \log_2 (2)^{-6} - \sum_{i=1}^{48} \left( \frac{1}{32} \right) \log_2 (2)^{-5} \]

\[ H(X) = -32 \times \left( \frac{1}{64} \right) \times (-6) - 16 \times \left( \frac{1}{32} \right) \times (-5) \]

\[ H(X) = 5.5 \]

The number of legitimate flows except SYN requests cannot increase more than valid number of connections for a server that uses TCP service at transport layer. Accordingly, keeping an account of valid number of connections a maximal constraint on sample entropy value can be enforced to detect any kind of anomaly.

![Figure 4.1: Maximum value of sample entropy](image-url)
Moreover, low rate attacks consist of a lot of traffic streams having lesser number of packets than legitimate streams in an observed sample of packets. This results in a negatively skewed distribution. Similarly, in high rate attacks, mostly attack streams contribute to more number of packets than legitimate traffic streams, which result in a positively skewed distribution. Sample entropy $H(X)$ for negatively skewed distribution is more whereas for positively skewed distribution its value is lesser than normal $H(X)$ without attack (Klugh, 1994). The facts indicate that sample entropy $H(X)$ is an effective summary statistic for characterizing a distribution and can be used to detect DDoS attacks.

4.2 Source Address and Traffic Cluster Entropy

Source address and traffic cluster based entropy have been proposed to detect DDoS attack as well as to discriminate it from FE. In both DDoS attack and FE, the increase in volume of traffic is far more than normal. However the way both should be tackled is totally different as discussed in section 2.4.2 of chapter 2. In case of FE, when some new event is launched on web site, suddenly request rate increases to access newly launched item. The clients generating requests can be grouped into clusters on the basis of network and subnet addresses. Krishnamurthy and Wang (2000) have defined a cluster as a group of clients that are close together topologically and are likely to be under a common administrative control. In our approach, we have formed clusters on the basis of 16bit and 24bit prefix of source IP addresses. In case of 16bit clusters, all source IP addresses that share common initial 16 bits are treated as single traffic cluster and in 24 bit clusters, all the source IP addresses that share common initial 24 bits are considered in same traffic cluster for the purpose of monitoring and analysing traffic. Since clusters sending a large number of requests to the server can be frequently observed during FE (Jung et al., 2002), the traffic clusters of requests received by the server during a FE are not random. So there will be minimal increase in number of clusters. On the other hand, the attackers always fabricate a lot of data packets, and the source addresses of these packets are generally different and randomly distributed.

The current approach is based on the fact that during a DDoS attack, distribution of source IP addresses as well as 16bit and 24 bit clusters tend to be
more dispersed than under normal conditions. However in case of FE, there is increase of legitimate request rate due to sudden increase in number of legitimate clients. So there is increase in dispersion of source IP address distribution, but as increase of clients is from same traffic clusters as of earlier, so dispersion in 16bit and 24 bit cluster distributions do not increase.

A time series dispersion analysis in distributions of source IP addresses, 16 bit and 24 bit traffic clusters are performed in the proposed work. The entropy is used as a summarizing metric for computing dispersion of various traffic distributions. Entropy not only captures degree of dispersal but also concentration of a traffic distribution. The increase in entropy signals increase in randomness of the distribution i.e. dispersion. The smaller the entropy, the narrower the distribution range of packets’ source addresses. Under normal network condition, the entropy of network packets always fluctuates in a narrow range. However when attacks come out, the entropy value of source IP address, 16 bit and 24 bit traffic clusters distributions have noticeable changes. On the other hand, the traffic in FEs has very less number of unique traffic clusters as compared to source addresses as observed by Jung et al. (2002). So, source IP address entropy is expected to increase whereas entropy of 16 bit traffic clusters as well as 24 bit traffic clusters shall not increase in case of FEs. So on the basis of source IP entropy and traffic cluster entropy, we can detect the DDoS attack as well as discriminate it from FE. The source IP address entropy and traffic cluster entropy are computed as below:

Let src_IP is the random variable observed in different packets which can take values from the set \{src_IP_1, src_IP_2, src_IP_3, ..., src_IP_n\}. Here n is total number of sources.

Let Δ seconds is the sample time for traffic monitoring.

In time window \( \{t - \Delta, t\} \), let number of packets received are \( X_i = \{X_{i1}, X_{i2}, X_{i3}, ..., X_{in}\} \) for corresponding src_IP from the set \{src_IP_1, src_IP_2, src_IP_3, ..., src_IP_n\}.

Then sample entropy of random variable src_IP given by \( H(src_IP) \) as per equation 4.1 in time window \( \{t - \Delta, t\} \) is given as below:-
\[ H(src\_IP) = - \sum_{i=1}^{n} p(src\_IP_i) \times \log_2 p(src\_IP_i) \]  

(4.2)

Here the probability of occurrence of each src\_IP i.e. \( p(src\_IP_i) \) is given as

\[ P(src\_IP) = \{ p(src\_IP_1), p(src\_IP_2), p(src\_IP_3), \ldots \} \]

where \( p(src\_IP_i) = \frac{X_i}{S} \) and

\[ S = \sum_{i=1}^{n} X_i \]

Let tc\_ID is the random variable representing traffic cluster ID derived from src\_IP observed in different packets.

Let \( tc\_ID \) random variable takes values from the set \( \{ tc\_ID_1, tc\_ID_2, tc\_ID_3, \ldots, tc\_ID_m \} \). Here \( m \) is the total number of clusters observed.

Let \( \Delta \) seconds is the sample time for traffic monitoring.

In time window \( [t - \Delta, t] \), let the number of packets received are \( Y_i = \{ Y_1, Y_2, Y_3, \ldots, Y_m \} \) for corresponding \( tc\_ID \) from the set \( \{ tc\_ID_1, tc\_ID_2, tc\_ID_3, \ldots, tc\_ID_m \} \).

Then sample entropy of random variable \( tc\_ID \) given by \( H(tc\_ID) \) is as below:

\[ H(tc\_ID) = - \sum_{i=1}^{m} p(tc\_ID_i) \times \log_2 p(tc\_ID_i) \]  

(4.3)

Here the probability of occurrence of each \( tc\_ID \) i.e. \( P(tc\_ID) \) is given as

\[ F(tc\_ID) = \{ p(tc\_ID_1), p(tc\_ID_2), p(tc\_ID_3), \ldots, p(tc\_ID_m) \} \]

where \( p(tc\_ID_i) = \frac{Y_i}{W} \) and

\[ W = \sum_{i=1}^{m} Y_i \]

Equation 4.2 is used to compute source IP address entropy. Equation 4.3 is used to compute entropy of 16 bit and 24 bit traffic clusters. In this approach, source
IP address, traffic cluster entropies are computed in a time window \( \{t - \Delta, t\} \). The computed source IP address, 16 bit and 24 bit traffic cluster entropies are compared with their respective thresholds. If there is no appreciable change in source IP address as well as 16 bit and 24 bit traffic cluster entropies, it signifies legitimate traffic. However if there is increase in source IP address entropy but 16 bit and 24 bit traffic cluster entropies do not increase from their thresholds, it signifies flash crowd or event. On the other hand, if there is increase in source IP address as well as 16 bit and 24 bit traffic cluster entropies, it indicates DDoS attack. Both DDoS attacks and flash crowd are required to be handled separately. In order to handle DDoS attacks, filters need to be enabled whereas, to handle FE, extra resources are deployed.

4.3 DDoS Defense System Modules

A typical DDoS defense system mainly consists of detection of attack, characterization of attack sources, and rate limiting or filtering of attack traffic. The process of identifying that a network or server is under attack after launch of attack is called detection. Detection requires monitoring of traffic and its sophisticated analysis. Source IP address entropy and traffic cluster entropy are used for detection in our approach. Characterization means differentiating attack traffic from legitimate traffic. Rate limiting and Filtration is used to mitigate DDoS traffic so that legitimate traffic should not suffer. Thus, a comprehensive DDoS solution requires five effective modules namely traffic monitoring, traffic analysis, detection of attack, characterization of attack traffic and attack traffic filtration (Mirkovic et al., 2004). System modules of DDoS defense are shown in figure 4.2.

Clearly, the first module i.e. traffic monitoring accepts attack and legitimate traffic as input and generates required traffic data structures. These traffic data structures are utilized by traffic analysis module for computation of detection metric. In this approach, source IP address entropy and traffic cluster entropy are computed. The computed detection metric is compared with the baseline detection metric in the next module i.e. detection of attack. If computed detection metric deviates from the baseline beyond a threshold, attack is said to be detected. Once the attack is detected, characterization module takes the control and segregate attack traffic from
legitimate traffic. The attack signatures generated by characterization module are used by attack traffic filtration module to mitigate or eliminate the effect of attack.

Figure 4.2: DDoS defense system modules

4.3.1 Traffic Monitoring

The current network systems can simply be divided into two domains. The first domain is the core network. A core network usually consists of high-speed core routers. It is the backbone network which is in charge of transferring traffic at fast rate. The edge network is another domain which connects to a core network through edge routers. An edge network may represent a single customer network or ISP domain. Within an ISP domain, same structure is there i.e. core routers are used for fast transmission of network traffic and edge routers are used for connecting to customer domains or for peering with other ISP domains. In this approach, traffic is monitored at edge routers of ISP domain. Packets are monitored on the basis of source IP address and traffic clusters. The terminology used is explained below: -
Source IP address (src_IP): A 4-byte logical address used in the packets to represent its source IP.

Traffic cluster (tc): The traffic generated from same networks, subnets or administrative domains is defined as traffic cluster.

16-bit traffic cluster identifier (tc16_ID): All the packets which share the same initial 16 bits of their src_IP are in same group called 16-bit traffic cluster. It is obtained by bit-wise AND operation of src_IP and 16-bit mask i.e. 255.255.0.0. A unique identifier assigned to such a traffic group or cluster is defined as 16-bit traffic cluster identifier. Formation of 16bit clusters from source address is shown in figure 4.3.

24-bit traffic cluster identifier (tc24_ID): All the packets which share the same initial 24 bits of their src_IP are in same group called 24-bit traffic cluster. It is obtained by bit-wise AND operation of src_IP and 24-bit mask i.e. 255.255.255.0. A unique identifier assigned to such a traffic group or cluster is defined as 24-bit traffic cluster identifier. Formation of 24bit clusters from source address is shown in figure 4.3.

Figure 4.3: Formation of clusters
Basically three linked lists, sourceipcluster, 16bitcluster and 24bitcluster are headers of the linked lists which are used to store source addresses, 16bit traffic clusters and 24bit traffic clusters respectively along with the count i.e. no of packets arrived for a particular source IP or cluster.

Packets are monitored in a short sized time window \( \{t-\Delta, t\} \) to minimize memory overheads as shown in figure 4.4. Here \( \Delta \) seconds is the size of time window. During sampling interval \( \Delta \) seconds, packets are monitored and different type of clusters is formed from observed source IP address of the packets. The flowcharts for monitoring of traffic and formation of clusters are shown in figure 4.5 and 4.6b respectively. At time \( t \), the monitoring process yields packets arrival distribution of src_IP and tc_ID. Then the probability of occurrence of each src_IP and tc_ID i.e. \( P(\text{src\_IP}) \), \( P(\text{tc16\_ID}) \) and \( P(\text{tc24\_ID}) \) are respectively computed and analysed further.

![Monitored Traffic](image)

**Figure 4.4: Sampling interval for attack detection and reaction**

4.3.2 Traffic Analysis using Source Address and Traffic Cluster Entropy

In this module, source address entropy \( H(\text{src\_IP}) \) and traffic cluster entropies \( H(\text{tc16\_ID}) \) and \( H(\text{tc24\_ID}) \) are computed using equation 4.2 and 4.3 respectively for the fixed size time window \( \{t-\Delta t\} \). As shown in figure 4.5, after every \( \Delta \) seconds, source address entropy and traffic cluster entropies are computed and variables TPA, sourceipcluster, 24bitcluster and 16bitcluster are set to NULL. The
flowchart for computation of source address entropy and traffic cluster entropy is shown in figure 4.6a.

![Flowchart for traffic monitoring and analysis](image)

Figure 4.5: Flowchart for traffic monitoring and analysis

105
4.3.3 Detection of Attack

Entropy summarises random distribution of features observed or derived from network traffic. These network features can be source IP address, 24 bit traffic cluster, 16 bit traffic clusters etc. The bigger the entropy, more random the corresponding feature is. The smaller the entropy, the narrower distribution range of the feature is, and some particular feature values may have high occurrence probability too. It has been found that the source IP address, 16bit and 24bit traffic clusters entropies fluctuate to some extent under normal network conditions. But during attack the entropy values have perceptible changes. The change in the feature distribution is detected through monitoring time series variation in the entropy.
In this work, an anomaly based detection approach is used. First, a legitimate traffic model is created keeping in view the network state in normal conditions. Mostly a network state summarising metric called detection metric is computed and a threshold of its value is fixed for the normal state. Then the detection metric is regularly computed on the actual state of the network and the comparison of this actual state is made with the threshold on the regular basis. The appreciable difference in current and baseline profiles signals an occurrence of unusual event in the network. The unusual event can be any anomaly like DDoS attack or flash traffic.

It has been found that without attack, source IP address entropy $H_{\text{src IP}}$ and traffic cluster entropy $H_{\text{tc ID}}$ values vary within narrow limits after slow start phase is over. This variation becomes narrower if we increase the monitoring period. In the proposed approach, $H_{\text{src IP}}$ and $H_{\text{tc ID}}$ are used as detection metric. $H_{\text{src IP}}$ and $H_{\text{tc ID}}$ are computed after every $\Delta$ seconds in normal state. Average values of $H_{\text{src IP}}$ and $H_{\text{tc ID}}$ are designated as normal source IP address entropy $H_{N_{\text{src IP}}}$ and normal traffic cluster entropy $H_{N_{\text{tc ID}}}$. To detect the attack, the current source IP address entropy $H_{C_{\text{src IP}}}$ and current traffic cluster entropy $H_{C_{\text{tc ID}}}$ is calculated in shorter time window continuously. Whenever there is appreciable deviation of $H_{C_{\text{src IP}}}$ and $H_{C_{\text{tc ID}}}$ from $H_{N_{\text{src IP}}}$ and $H_{N_{\text{tc ID}}}$ respectively, attack is said to be detected. It is worth mention here that for computing $H_{N_{\text{src IP}}}$ and $H_{N_{\text{tc ID}}}$, as well as corresponding thresholds for the network environment, appropriate experimentations are performed so as to generate accurate results. These experiments are detailed in subsequent chapters.

The timeline for the network state is depicted in figure 4.7. It is assumed that the network is under heavy load at time $t_a$, which means that either attacking sources are emitting packets from this time (DDoS attack) or there is increase in legitimate request rate (FE). Both events require different tackling mechanisms. In case of FE, extra resources are to be provided to give service to extra load. But in case of attack, filtration of extra attack traffic is required. So, it is very important to first detect the unusual event and then to designate the same to either DDoS attack or flash event.
Clearly, from the timeline depicted in figure 4.7, the network is in normal state for
time $t < t_d$ and turns into heavy load state at time $t_d$. Let $t_d$ denote our estimate on $t_d$
At time $t_d$ we have to distinguish between two events mainly. Flowchart for
detection of attack is given in figure 4.8. Following events can trigger at time $t_d$

**Case I:**

$$H_C(\text{src\_IP}) > (H_N(\text{src\_IP}) + a \cdot d_{\text{src\_IP}})) \text{ AND } (H_C(t_c\_ID) > (H_N(t_c\_ID) + z \cdot d_{t_c\_ID}))$$

status = DDoS\_attack;  

**Case II:**

$$H_C(\text{src\_IP}) > (H_N(\text{src\_IP}) + a \cdot d_{\text{src\_IP}})) \text{ AND } (H_C(t_c\_ID) \equiv (H_N(t_c\_ID) + z \cdot d_{t_c\_ID}))$$

status = FE

**Case III:**

$$H_C(\text{src\_IP}) > (H_N(\text{src\_IP}) + a \cdot d_{\text{src\_IP}})) \text{ AND } (H_C(t_c\_ID) < (H_N(t_c\_ID) - z \cdot d_{t_c\_ID}))$$

status = SPDDoS\_attack  // Subnet spoofed DDoS attack

**Case IV:**

$$H_C(\text{src\_IP}) < (H_N(\text{src\_IP}) - a \cdot d_{\text{src\_IP}}))$$

status = HSDDoS\_attack  // High rate skewed DDoS attack

**Case V:**

$$H_C(\text{src\_IP}) \equiv (H_N(\text{src\_IP}) + a \cdot d_{\text{src\_IP}})) \text{ AND } (H_C(t_c\_ID) > H_N(t_c\_ID) + z \cdot d_{t_c\_ID}))$$

status = SOPDDoS\_attack  //Sophisticated DDoS attack;

Here $a, z \in \mathbb{Z}$ where $I$ is set of integers. Tolerance factor $a$ and $z$ are design
parameter and $d_{\text{src\_IP}}$ is standard deviation in source IP address entropy, and $d_{t_c\_ID}$ is standard deviation in traffic cluster entropy computed while profiling
normal behaviour for the network without attack. Flowchart in figure 4.8 illustrates
the complete detection mechanism. In scenario 1, as per equation 4.4, if $H_C(\text{src\_IP})$
and $H_C(t_c\_ID)$exceeds their thresholds, then it is a sign of DDoS attack. DDoS
attacks make use of millions of zombies scattered all around the Internet. Clearly
when the scrupulous traffic from these zombie machines reach in the victim ISP
domain, it abruptly increases $H_C(\text{src\_IP})$ and as traffic is coming from multiple
networks so $H_C(t_c\_ID)$ also increases. In equation 4.5, if $H_C(\text{src\_IP})$ exceeds its
threshold but $H_c(tc\_ID)$ remains in congruence with threshold limits, it signifies FE. As per Jung et al. (2002), a lot of new sources send requests towards victim server but the number of new unique clusters is very less which results in an appreciable increase of $H_c(src\_IP)$ but negligible variation in $H_c(tc\_ID)$. On the other hand, as per equation 4.6, if $H_c(src\_IP)$ exceeds its thresholds but $H_c(tc\_ID)$ decreases below its thresholds, it signifies subnet spoofed DDoS attack. In this scenario, increase in number of sources results in increase of $H_c(src\_IP)$ but these sources are from same networks which give a sign of spoofing within the same network. It results in increased number of packets for some specific traffic clusters i.e. distribution of traffic among traffic clusters is skewed in few clusters only. Due to this skewness, traffic cluster entropy decreases below the threshold. If $H_c(src\_IP)$ decreases than its threshold as per equation 4.7, then it means high volume DDoS attack. This scenario depicts the situation in which a few sources generate large amount of traffic towards the victim server. So traffic distribution among different source IP addresses is skewed in few IP addresses only. Hence, instead of increase in $H_c(src\_IP)$, there is appreciable decrease in $H_c(src\_IP)$ . In the last scenario, if $H_c(src\_IP)$ remains in congruence with the threshold limits and $H_c(tc\_ID)$ exceeds its thresholds, it signifies a sophisticated DDoS attack. Sophisticated DDoS attacks do not involve many zombie machines from the same networks so to avoid detection. Moreover if number of sources also does increase much then $H_c(src\_IP)$ also remains in limits. Then attack is very difficult to detect but in this case attack to have some impact zombie machines have to be used from multiple networks. The use of multiple networks may result in increase in $H_c(tc\_ID)$. Hence the attack can be trapped. Clearly if attacker knows the threshold limits of $H_c(src\_IP)$, and is keeping an eye on the network also then this scenario can prove to be helpful.

![Figure 4.7: Timeline of network state](image)

109
Figure 4.8: Flowchart for detection of DDoS attack and flash event

Many detection systems use fixed thresholds so as to suit their approach without following appropriate threshold calibration methods (Carl et al., 2006). This method makes sense for some applications. For example, in control systems, there may be a certain tolerance level for products to be considered “acceptable”. If this tolerance level is exceeded, then the product is considered “bad”. But for network traffic monitoring, the background is continually changing. If the background traffic changes, these thresholds may become meaningless and need to be changed. If the threshold $z$ is set too high, normal entropy varies from $H_N(X) - z \cdot d$ to $H_N(X) + z \cdot d$. 

110
This range of normal entropy that classifies traffic as legitimate would be broad, so false positive rate will be low, but detection rate will be low too. Similarly, if the threshold \( z \) is set too low, it may detect most attacks, but suffer from a high false positive rate. Therefore, in network traffic monitoring, it is critical to update these estimates adaptively. Adaptive thresholds have been used in this work, which means that the threshold is updated regularly depending upon network conditions and user requirements. Since this adaptive approach continually updates the threshold or a tolerance factor, the model adjusts to reflect changes in background traffic. False positive gives the effectiveness of the approach whereas false negative gives a measure of system reliability. Variations in thresholds or tolerance factor quantifies false positive and false negative. Minimization of false positives and false negatives assist in making decision on the optimum value of \( z \). By adjusting a baseline, estimates adjust quickly to calibrate the system for normal entropy range.

4.3.4 Characterization of Attack Sources or Traffic

Once the attack has been successfully detected, the next crucial task is attack characterization. The characterization module must be able to precisely describe the offending traffic, so that it can be sifted from the rest by the response module. Legitimate and attack traffic models used in detection, sometimes coupled with additional statistics and profiling, guide the attack characterization. The goal is to obtain a list of parameters from the packet header and contents, along with a range of values that indicate a legitimate or an attack packet. Each incoming packet is then matched against the list, and the response is selectively applied to packets deemed to be a likely part of an attack. Attack characterization is severely hindered by the fact that the attack and legitimate traffic look alike. However, good attack characterization is of immense importance to DDoS defense, as it determines the amount of collateral damage and the effectiveness of the response.

Characterization of attack traffic is based on the fact that during DDoS attack, most of the traffic clusters are new i.e. not encountered earlier. On the other hand, during normal operation of the network, most of the source addresses repeat. In order to validate the rational that the IP source addresses to a network are consistent during normal operation, Peng (2007) analysed daily IP Address Database
Design traces of Auckland. He found that about 88-90% of IP addresses that appear in the network under normal conditions have previously appeared in the network. A similar study was also done on Small ISP Trace, and it was found that about 80% of IP addresses appeared in the last two weeks repeated in the next week. Two weeks time was chosen as the history period in these studies. Obviously, with an increase in the length of the history period, the percentage of recurring IP addresses also increases. Peng (2007) used this idea for detection of attacks, however the similar rational can be used for the purpose of characterisation of attack sources, after detection of attack. It is a known fact that during FEs, most of the flash traffic originates from those networks from where already some traffic has been recently sent to the server. So to distinguish FE traffic from DDoS, finding new traffic clusters can prove to be helpful. On the basis of this fact, two fixed size matrices namely history based traffic cluster matrix (TCMH) and current window based traffic cluster matrix (TCMC) are defined. These matrices comprise of 256*256 elements. TCMH is used to maintain history whereas TCMC is used to store timestamp of packets received in current time window. Matrix construction is performed to capture the cluster distribution of incoming packets. The construction of matrices is made as simple as possible, so that it can be implemented on high-speed routers. The matrix is constructed using following method:

Each IP address is divided into four octets as presented in the following notation, where the length of each octet is one byte.

IP1.IP2.IP3.IP4

In this technique, we simply map an incoming packet to a specific location in the matrix determined by IP2 (used as the row index of the matrix) and IP3 (used as the column index of the matrix) of the IP address.

\[ i = \text{IP2} \text{ and } j = \text{IP3} \]

Here i is the row index and j is the column index of the matrix. A very nice property of this simple approach is that it results in a matrix of fixed size, which is a property that many other clustering mechanisms do not have.
Figure 4.9 illustrates how an entry of the matrix is located based on the IP address of the incoming packet. The matrix is initialized with zeros in all entries.

**Figure 4.9: Indexing incoming packet into matrix in current time window**

1. \( \text{timestamp}() \)
2. \( X=Y=0 \)
3. \( \text{seconds} = \text{gettime}() \)
4. If \( \text{seconds} \geq 86400 \)
   - No
   - \( X = \text{seconds} \)
   - \( Y = \text{seconds} \mod 86400 \)
   - \( \text{timestamp} = X, Y \)
   - Return \( \text{timestamp} \)
5. Yes
   - \( X = \text{seconds} \)
   - \( Y = \text{seconds} \mod 86400 \)
   - \( \text{timestamp} = X, Y \)
   - Return \( \text{timestamp} \)

**Figure 4.10: Flowchart for computing timestamp**
For each incoming packet, the matrix entry based on the IP address is located and initialized with recent time stamp of the packet. The method for computing recent timestamp is given in figure 4.10. The structure of the computed timestamp is X.Y where X represents number of days and Y represents seconds w.r.t. fixed date and time. Elapsed time with reference to a fixed date and time is taken in tseconds variable. Since the number of seconds in one day are 24*60*60=86400, so if tseconds are less than 86400, timestamp is set as 0.tseconds, otherwise number of days are computed in X, and Y gives number of seconds of last day.

Flowchart in figure 4.11 outlines the method of matrix construction. TCMC denote the matrix constructed by processing incoming packets during the Δ time unit. After expiry of current time window, matrix TCMH (History based traffic cluster matrix) is updated from TCMC. History information of packets received in last few days is maintained. Number of days for which history is to be maintained is a tuneable parameter that depends on the load of traffic. In this approach 14 days have been chosen for maintaining the history. Everyday i.e. after 86400 seconds, the age of each entry in TCMC is computed. If age is more than 14 days then corresponding entry is set to zero. Moreover entries are further refined by keeping track of response packets. Those entries for which response packets are not seen within a stipulated time, are also set to zero. A typical snapshot of the history matrix TCMH is shown in figure 4.12.

Two matrices BTCMC (Bit wise current window based traffic cluster matrix) and BTCMH (Bit wise history based traffic cluster matrix) of 256 X 256 are defined and initialized to 0. Characterization module is activated only when detection module raises an alert of any kind of DDoS attack. Let attack is detected at time t_a. All nonzero entries of TCMC are set to 1 in BTCMC (Bit wise current window based traffic cluster matrix) and nonzero entries of TCMH are set to 1 in BTCMH. BTCMA is another matrix in which each entry is calculated as XOR (exclusive-or) of the corresponding entries in BTCMH and BTCMC. The XOR operation between BTCMH and BTCMC as shown in figure 4.13 is used to remove the overlapping
Symbols used

TCMH[256][256] -历史基于流量簇的矩阵，用于存储src_IP的第2和第3字节对应的时戳

TCMC[256][256] - 流量簇矩阵，用于存储当前时间窗口内src_IP的第2和第3字节对应的时戳

\(\Delta\) - 窗口大小，取为0.5秒

timestamp - 从参考日期和时间计算出的经过时间

separate_octets(src_IP) - 函数，用于分离IP地址的各个字节，这些字节存储在数组oct[4]

图4.11: 构建TCMC和TCMH矩阵流程图

图4.12: TCMH矩阵快照
clusters, since \( 1 \oplus 1 = 0 \). BTCMA has entry 1 for new clusters. Since new clusters are responsible for DDoS attack traffic, thus, all non zero entries of BTCMA refers to the attack signature. Attack signatures can be printed as “*i,j*” \( \forall i, j \) where BTCMA[i][j] \( \neq 0 \).

\[
\begin{array}{cc}
\text{BTCMA} & = \text{BTCMH XOR BTCMC} \\
\hline
0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 \\
1 & 0 & 0 & 1 \\
\hline
\text{BTCMH} & \\
1 & 1 & 1 & 0 \\
1 & 1 & 1 & 0 \\
0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
\hline
\text{BTCMC} & \\
1 & 1 & 0 & 1 \\
1 & 1 & 1 & 1 \\
0 & 0 & 1 & 1 \\
1 & 0 & 1 & 1 \\
1 & 1 & 0 & 1 \\
\end{array}
\]

**Figure 4.13: XOR operation between matrices BTCMH and BTCMC**

TCMH is updated from TCMC after every \( \Delta \) seconds. Once attack is detected, next important task is characterization i.e. to identify attack traffic. Flowchart for characterization of attack traffic is given in figure 4.14. Characterization of attack traffic is performed by identifying new traffic clusters.

The proposed characterization approach however is susceptible to attacks in which a sneaky attacker falsely trains our history to include attack clusters. An attacker can slowly send attack traffic from legitimate domains if he is able to exploit the vulnerabilities of the hosts present in that domain. Though, it is difficult preposition. But we need to think a step ahead. Once exploited, these hosts can also be used as zombies to launch attack against the victim. In that case, the proposed approach will be able to detect the attack, but will not be able to filter attack traffic thus will result in collateral damage. However, the proposed characterization approach can be extended to overcome this limitation. The deviation in entropy can provide an estimate of number of attack domains used to launch the attack by using regression (kumar, 2008) and multiple regression (Gupta et al., 2010, 2011). This
estimate can act as threshold to find number of attack clusters involved in launching the attack. If the number of attack clusters computed by proposed approach is less than the threshold, then one day before history based traffic cluster matrix (TCMH) can be compared with current window based traffic cluster matrix (TCMC) to find suspected domains. However if more than half of traffic clusters in history are picked, then it should be treated as a sign that the history is compromised by the attackers, so manual intervention is required to relieve the victim from the attack.

**Symbols used**

TCMH[256][256] History based traffic cluster matrix to store timestamp corresponding to 2<sup>16</sup> and 3<sup>16</sup> octet of src_IP

TCMC[256][256] Traffic cluster matrix to store timestamp of packets in current time window corresponding to 2<sup>16</sup> and 3<sup>16</sup> octet of src_IP

BTCMH[256][256] Bitwise history based traffic cluster matrix with 1-bit elements. All nonzero entries of TCDH are stored as bit 1 and all zero entries are set to zero in BTCMH matrix

BTCMC[256][256] Bitwise current window based traffic cluster matrix with 1-oct element. All nonzero entries of TCMC are stored as bit 1 and all zero entries are set to zero in BTCMC matrix

Attack_signature An array to store signatures of attack traffic clusters

**Charaterisation(TCMH,TCMC)**

Set all nonzero entries of TCMH to 1 in matrix BTCMH

Set all nonzero entries of TCMC to 1 in matrix BTCMC

BTCMA=BTCMH XOR BTCMC

set i=0, j=0,

s=0

No

if (j <= 255)

Return Attack_signature

Yes

i=i+1

if (j <= 255)

Yes

if BTCMA[i][j]=1

No

Yes

Attack_signature[eq. "x..i..j.."]

s=s+1

j=j+1

Figure 4.14: Flowchart for characterization of attack traffic
4.3.5 Filtration of Attack Traffic

A planned reaction against DDoS attacks called response includes filtration of attack traffic as close as possible to the origin of mischievous traffic. The flowchart for filtration module is given in figure 4.15.

In this approach, attack traffic filters are deployed at ingress points called Point of Presence PoPs of the protected ISP domain. The characterization module discussed in the last section works at PoP P, connected to the protected server. All the PoPs of the protected domain share the same multicast group. Attack signatures found in characterization module at PoP P, are communicated to all the PoPs of the protected ISP domain through multicasting. The filter array at all the PoPs are populated with attack signatures. Traffic entering in the ISP domain from any PoP is checked against the corresponding filter array. If source IP address is characterized in the filter array, then the packet is dropped, otherwise the packet is allowed to pass. Keeping in view overhead of filtering, a statistically set threshold has been set, which helps in setting the filter on or off. The multicast communication framework used in this approach is itself not secure and is prone to security attacks. Moreover, this communication framework can also be extended in an incremental fashion among multiple cooperative ISP domains. Both of these issues are included in our future work.

4.4 Motivation for DDoS defense in ISP Domain

The DDoS threat can be countered at different locations in the network (Mirkovic et al., 2004). A DDoS attack consists of several streams of attack packets originating at different source networks. These traffic streams pass through various ISPs, in the core of the Internet called intermediate network and reach up to the victim. Figure 4.16 depicts this interaction among source networks, intermediate network and victim network. Defense mechanisms can be placed at some or all of these locations. Each of the involved networks i.e. source, intermediate, and victim can host DDoS defense systems. A feasibility analysis of DDoS defense deployment is done at each of these individual points. Each location has its strengths and weaknesses. A detailed analysis of DDoS defense deployment at possible locations is given below.
Figure 4.15: Flowchart for filtration of attack traffic
4.4.1 Victim Network

The most obvious location for a DDoS defense system is near the victim. Defense could be located on the victim's own machine, or at a router, firewall, gateway, proxy, or other machine that is very close to it. Most existing defense mechanisms that protect against other network threats tend to be located near the victim, for very good reasons. Many of those reasons are equally applicable to DDoS defense also. Nodes near the victim are in good positions to know when an attack is ongoing. They are able to directly observe the attack, but even if they cannot, they are quite close to the victim and often have a trust relationship with that victim. The victim can easily send them alerts, when it is under attack. Also, the victim is the single node in the network that receives the complete information about the characteristics of the attack, since all of the attack packets are observed there. Mechanisms located elsewhere see only a partial picture and might need to take action based on incomplete knowledge.

Another advantage of locating a defense near the target is deployment motivation. Those who are particularly worried about the danger of DDoS attacks
will pay the price of deploying such a defense mechanism, while those who are unaware or does not care about the threat need not pay. Further, the benefit of deploying the mechanism accrues directly to the entity that paid for it. Historically, mechanisms with these characteristics (such as firewalls and intrusion detection systems) have proved to be more widely accepted than mechanisms that require wide deployment for the common good (such as ingress/egress filtering of spoofed IP packets).

A further advantage of deployment near the target is maximum control by the entity receiving protection. If the defense mechanism proves to be flawed, perhaps generating large numbers of false positives, the target machine that suffers from those flaws can turn off or adjust the defense mechanism fairly easily. Similarly, different users who choose different trade-offs between the price they pay for defense and the amount of protection they receive can independently implement those choices when the defense mechanisms are close to them and are under their control.

But there are also serious disadvantages to defense mechanisms located close to the victim. A major disadvantage is that a DDoS attack, by definition, overwhelms the target with its volume. Unless the defense mechanism can handle this load more cheaply than the target, or is much better provisioned than the target, it is in danger of being similarly overloaded. Instead of spending a great deal of money to heavily provision a defense box whose only benefit is to help out during DDoS attacks, one might be better off spending the same money to increase the power of the target machine itself. In some cases where the defense mechanism is just a little bit upstream of the potential target, we can gain advantages by sharing the defense mechanism among many different potential targets, somewhat lessening this problem, since several entities can pool the resources they are willing to devote to DDoS defense on a more powerful mechanism.

A less obvious problem with this location is that the target may be in a poor position to perform actions that require complex analysis and differentiation of legitimate and attack packets. The defense mechanism in this location is, as noted above, itself in danger of being overwhelmed. Unless it is very heavily provisioned,
it will need to perform rather limited per-packet analysis to differentiate good
packets from attack traffic. Such mechanisms are thus at risk of throwing away the
good packets with the bad.

A further potential disadvantage is that, unless the solution is totally
automated and completely effective, some human being at the target will have to
help in the analysis and defense deployment. If there is no person capable of doing
that, assistance of others who are not at the site is to be taken, which limits the
advantages of the defense being purely local. Further, if the flood is large and the
necessary countermeasures are not obvious, many of the local resources could well
be overwhelmed. This problem may not be too serious for very large sites that
maintain several highly trained systems and network administrators, but it could be
critical for a small site that has few or no trained computer professionals on its
regular staff.

A final disadvantage is that deployment near each potential target benefits
only that target. Every edge network that needs protection must independently
deploy its own defense, gaining little benefit from any defense deployed by other
edge networks. The overall cost of protecting all nodes in the Internet using this
pattern of deployment might prove higher than the costs of deploying mechanisms at
other locations that provide protection to wider groups of nodes.

4.4.2 Source Network

Source end defense is to deploy a defense mechanism near attack sources.
Such a defense could be statically deployed at most or all locations from which
attacks could possibly originate or could be dynamically created at locations close to
where streams belonging to a particular ongoing attack actually are occurring. One
advantage of this deployment location is that DDoS attack streams are not highly
aggregated close to the source, unlike close to the attack's target. They are of a much
lower volume, allowing more processing to be devoted to detecting and
characterizing them than is possible close to the target. This low volume and lack of
aggregation may also prove helpful in separating the packets participating in an
attack from those that are innocent traffic.
Prevention methods, such as Ingress/Egress Filtering (Ferguson and Senie, 1998) and repairing security holes (Geng and Whinston, 2002) are implemented at source networks to stop origin of DDoS traffic. Absence of incentives, per packet filtering overheads, and security measures awareness stand in the way of DDoS defense deployed at the source network. D-WARD (Mirkovic, 2003) is also a source-end defense scheme. It faces two hard challenges. First, in a highly distributed attack (i.e. isotropic DDoS attack), each source network is responsible for only a small fraction of the attack traffic, which is unlikely to generate anomalous statistics. Secondly, a witty DDoS attacker can also control the attack traffic from each source network to be within normal range because ultimately it is the aggregation of attack traffic and not individual source traffic which is going to inflict damage to the victim. Moreover, the biggest problem in source-end defense is requirement of global deployment which is impossible to achieve as Internet has no central control. The deployment scale required for this approach to be effective is very large. If attack streams emanate from 10,000 sources to converge on one poor victim, this style of defense mechanism would need to be deployed close to a significant fraction of those 10,000 sources to do much good. A DDoS defense mechanism that is only applied to 5 to 10% of the attack packets can do no good. The attacker would merely need to recruit 5 to 10% more machines to perform his attack, which is not a very challenging task. Unless the defense mechanism is located near a large fraction of all possible sites, it can not have enough coverage to be effective. Moreover motivation for source deployment is also low because it is unclear who would pay the expenses associated with this service.

4.4.3 Intermediate Network

Deployments in the intermediate network generally refer to defenses living at core Internet routers. As a rule, such defenses are deployed at more than one core router. For true core deployments, there are obvious advantages and disadvantages.

The vast bulk of the Internet's traffic goes through a relatively small number of core Autonomous Systems (ASs), each of which deploys a large, but not immense, number of routers to carry that traffic. Thus, any defense located at a reasonably large number of well-chosen ASs can get excellent coverage. So this
defense is effective, as it can provide its benefits to practically every node attached to the Internet.

There are some disadvantages of deploying DDoS defense in the middle of the network also. First, routers at core ASs are very busy machines. They cannot devote substantial resources to handle or analyse individual packets. Thus, a core defense mechanism can perform neither most cursory per-packet inspection nor any serious packet-level analysis to determine the presence, characteristics, or origins of a DDoS attack.

The basic problem in DDoS defense is separating the huge volume of DDoS traffic from the relatively tiny volume of legitimate traffic. DDoS defense at core routers cannot afford to devote adequate resources for making differentiation decisions. The core routers must have simple and computationally cheap rules for dealing with the vast majority of the packets they forward as per true Internet architecture.

A second problem arises because core routers can inflict massive collateral damage if they are not exceptionally accurate in discriminating DDoS traffic from legitimate traffic. If they make mistakes at a rate that might be acceptable for a victim-side deployment, they could easily drop a huge amount of legitimate traffic. Those running core routers consider dropping legitimate traffic as extremely undesirable. Combined with their lack of resources to perform careful examination of packets, it is always expected that the core defense should provide high accuracy with little analysis, which is indeed a very challenging task.

Another problem with this deployment location is that core routers are unlikely to notice DDoS attacks. They themselves are unlikely to be overwhelmed, and they cannot afford to keep statistics on packets coming through on a per-destination basis. Perhaps they can afford to look for DDoS attacks by a statistical method that examines a tiny fraction of the total packets, looking for suspiciously high numbers of packets to a single destination. But, it is clear that one node's overwhelming DDoS attack is another node's ordinary daily business. There is ongoing research on using measurements of entropy in packet traffic to detect DDoS attacks in the core. However, proven methods applied at core routers are not likely
to pinpoint all DDoS attacks without generating unacceptable levels of false positives.

Deployment incentives are also problematic for core-located DDoS defense mechanisms. By and large, the routers comprising the Internet backbone are not likely targets of DDoS attacks. They are meant for fast movement of packets from one end to another end. Thus, the companies running these machines do not probably receive direct benefit from deploying DDoS defense. They receive indirect benefit, since they typically try to minimize the time a packet travels through their system and because their business ultimately depends on the usability of the Internet as a whole. On the other hand, their equipment is expensive and must operate correctly even under conditions of heavy strain, so they are generally little inclined to install unproven hardware and software. If a core router defense performs badly, many users can be affected. Yet, unlike defense located in their own domains (whether source-side or victim-side), users have no power to turn the defense mechanisms off or adjust them. Moreover, those running the Internet backbone cannot afford to field calls from every ISP or, worse every user who is having his/her legitimate packets dropped by a core-deployed DDoS defense mechanism.

A final point against this form of defensive deployment is based on the respected end-to-end argument, which states that network functionality should be deployed at the endpoints of a network connection, not at nodes in the middle, unless it cannot be achieved at the endpoints or is so ubiquitously required by all traffic that it clearly belongs in the middle. While the end-to-end argument should not be regarded as the final deciding word in any discussion of network functionality, its careful application is arguably an important factor in the success of the Internet. Core-deployed DDoS defense mechanisms can run counter the end-to-end argument, provided one can make a strong case for the impossibility of achieving similar results at the endpoints. A comparison of various deployment locations with advantages, disadvantages and challenges is given in table 4.2.
<table>
<thead>
<tr>
<th>Deployment</th>
<th>Main features</th>
<th>Advantages</th>
<th>Disadvantages</th>
<th>Technical challenges</th>
</tr>
</thead>
<tbody>
<tr>
<td>Victim Network</td>
<td>Used to protect a set of hosts from being Attacked</td>
<td>• Most suitable for victim as it has to suffer losses due to attacks</td>
<td>• Computationally expensive due to high volume of traffic.</td>
<td>• Protection of legitimate traffic</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• DDoS attacks are easily detected due to huge volume of traffic</td>
<td>• Sometimes defense itself is vulnerable to DDoS attack due to high volume of traffic.</td>
<td>• Generating automatic attack alerts</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Deployment cost is low</td>
<td>• Filtering attack traffic is computationally expensive.</td>
<td></td>
</tr>
<tr>
<td>Source Network</td>
<td>Both detection and defeating components are deployed at the source end</td>
<td>• Filter attack traffic before it reaches target</td>
<td>• Very difficult to deploy as all networks cannot deploy unless enforced by legislation.</td>
<td>• How to detect an attack at the source without traffic aggregation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Detects attacks as soon as possible</td>
<td>• Lack of coordination</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Avoids overall network congestion as it stops attack traffic from polluting the entire Internet</td>
<td>• Less sensitive to catch attack signals</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Computation requirements of this solution is low</td>
<td>• ISP’s need to be financially motivated</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Low vulnerability</td>
<td>• Many deployment points</td>
<td></td>
</tr>
<tr>
<td>Intermediate network</td>
<td>A set of detection and response mechanisms are deployed in the core of Internet</td>
<td>• Better infrastructure available for deploying detection sensors and attack traffic filtering components</td>
<td>• Possible performance degradation</td>
<td>• Secure communication among defense modules</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Inter domain politics of isolation</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Attack detection is hard</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.2: Comparison of various deployment locations
Moreover viability of various DDoS defense modules with their vulnerability and relative deployment possibility are also explored in table 4.3 at all possible location. It had been realised by commercial organisation and researchers that distributed defense techniques are likely to be the proper solution for handling the DDoS threat (Robinson et. al., 2003). As source-side non distributed deployments just will not happen at a high enough rates to solve the problem. Victim-side deployments cannot handle high-volume flooding attacks. There is no single location in the network core where one can capture all attacks, since not all packets pass any single point in the Internet. Infrastructural solutions which span multiple networks and administrative domains and represent major undertakings of many Internet participants, are difficult to deploy and maintain. Further, the required cooperation of defenses is hard to achieve due to distributed Internet management and strictly autonomous operation of administrative domains. Securing and authenticating the communication channels also incurs a high cost if the number of participants is large.

### Table 4.3: Viability of DDoS defense at different deployment locations

<table>
<thead>
<tr>
<th>Deployment</th>
<th>Detection/characterization</th>
<th>Rate Limiting/Filtering</th>
<th>Defense Vulnerability/Robustness</th>
<th>Deployment Difficulty</th>
</tr>
</thead>
<tbody>
<tr>
<td>Victim Network</td>
<td>Easy</td>
<td>Difficult</td>
<td>High</td>
<td>Very Easy</td>
</tr>
<tr>
<td>Source Network</td>
<td>Very difficult</td>
<td>Easy</td>
<td>Low</td>
<td>Highly difficult</td>
</tr>
<tr>
<td>Intermediate Network</td>
<td>Difficult</td>
<td>Difficult</td>
<td>Medium</td>
<td>Difficult</td>
</tr>
</tbody>
</table>

In light of above said issues and Internet design vulnerabilities (Mirkovic and Reiher, 2004) a practical DDoS defense system deployment should have following important characteristics:

- Autonomous system i.e. whole defense location under one administrative control so that different defense nodes can collaborate in a secure manner.
- Large and infrastructure wise rich enough to handle high voluminous traffic from evenly distributed flood sources.
- Capability to evolve DDoS defense in incremental fashion.
- Sufficient financial motivation for value-added DDoS security service.
The Internet consists of thousands of Autonomous Systems (ASes) i.e., networks that are each owned and operated by a single institution. Usually each ISP operates one AS, though some ISPs may operate multiple ASes for business reasons (e.g. to provide more autonomy to administrators of an ISP’s backbones in the United States and Europe) or historical reasons (e.g. a recent merger of two ISPs) (Caesar and Rexford, 2005). An ISP has total autonomy to collaborate defense nodes in a secure manner. Enough infrastructures can be provided for DDoS defense to handle high volume at ingress points. Moreover, once agreement is reached between various ISPs then inter co-operation among ISPs is also possible (Chen and Song, 1993 and Tupakula and Varadharajan, 2003c). Accordingly, there is scope of incremental DDoS defense. If a provider’s infrastructure is attacked (routers, DNS, etc.), all services to its customers fail, resulting in service level agreement (SLA) violations. Moreover, ISPs normally host most of the services available on the Internet. The cost of DDoS protection is insurance against catastrophic failures that would cost the business orders of magnitude more in terms of both revenue and negative customer relations. However, Cost-avoidance is not the only motivation to implement a complete DDoS solution in ISP domain. DDoS protection can also be offered as value-added services to the users that create new revenue streams and provides competitive differentiation for ISPs. In nutshell, ISP level DDoS defense is most practical and viable at this stage. Though, longer term objective “how to achieve inter ISPs cooperation” still remains as the biggest challenge.

4.5 Abstract System Topology

The actual topology used for validation of proposed approach is shown in figure 5.5 of chapter 5. In order to explain the proposed approach, we have developed a system topology as shown in figure 4.17. Three ISP are shown and ISP1 is the protected ISP domain. ISPs contain many PoPs. These PoPs actually consist of interconnected edge and core routers. PoPs are connected to customer domains via edge routers and are attached with each other through high bandwidth links between their core routers. Moreover ISPs are joined with each through peering via their PoPs (Kumar, 2008). So these PoPs are entry and exit points of the ISPs. The legitimate and attack traffic from ISP1, ISP2 and ISP3 are directed towards the web server. Some of the customer domains have attack zombies. So customer domains
generate legitimate, attack, or legitimate and attack traffic towards the web server. Through peering points legitimate and attack traffic from other ISPs enter protected ISP$_1$.

4.6 Modular Architecture

The modular architecture of protected ISP$_1$ is shown in figure 4.18. It has mainly two parts, all the PoPs of the protected ISP domain and PoP P$_5$ through which service is provided to the ISP. At all PoPs of the protected ISP domain, traffic monitoring, entropy computation and traffic filtration modules are deployed. Traffic monitoring modules at PoPs of protected ISP monitors all the packets directed towards the web server. Entropy of all incoming packets is computed at PoPs of protected ISP after each time window $\{t-\Delta, t\}$. The computed entropies are then sent to the PoP P$_5$. At PoP P$_5$, cumulative entropy of all collected entropies is computed and compared with baseline behaviour and accordingly attack alert or flash alert is generated. If attack alert is generated, then characterization module identifies attack traffic and sends attack signatures to all the PoPs of protected ISP domain for traffic filtration. At all PoPs of the protected ISP, traffic filtration modules are deployed to filter attack traffic as per the attack signature.

![Figure 4.17: Abstract system topology](image-url)
4.7 Distributed Framework for DDoS Defense

The protected ISP in the distributed framework shown in figure 4.19, clearly indicates that at all the PoPs, traffic monitoring module is run, which not only separates source addresses from incoming packets but also classifies them into 16-bit and 24-bit traffic clusters. A time series analysis of this traffic is carried out at each PoP and computed entropy with total count of packets are sent to PoP $P_S$ connected to the server. The cumulative entropy is computed using equation 4.9 and 4.10 at PoP $P_S$.

$$H_s(src/IP) = (1/S_i) \sum_{i=1}^{P} S_i (H_s(src/IP) - \log(S_i)) + \log(S_i) \quad (4.9)$$

$$H_s(tc/IP) = (1/S_i) \sum_{i=1}^{P} S_i (H_s(tc/IP) - \log(S_i)) + \log(S_i) \quad (4.10)$$

Kumar (2008) has proposed equation 4.9 and proved analytically that if flows entering in PoPs are mutually exclusive then cumulative entropy is same as that of actual entropy computed at PoP $P_S$ for whole of the traffic. Here anomaly based detection module runs and checks for presence of attacks using cumulative entropy. The steps followed in distributed framework are as follows:-

Step 1. Source IP address is detached from the incoming packet destined to protected web server at all the PoPs except at PoP $P_S$.

Step 2. Classification into 16-bit and 24-bit cluster is done at each PoP by using bit-wise AND operation of each source IP address with 255.255.0.0 and 255.255.255.0 respectively.

Step 3. A count is maintained for each source IP, 16-bit, and 24-bit cluster in a time window $[t - \Delta_i, t]$.

Step 4. At the end of $\Delta_i$ seconds, source IP $H_s(src/IP)$, 16-bit traffic cluster $H_s(tc_{16}/ID)$ and 24-bit traffic cluster $H_s(tc_{24}/ID)$ entropies are computed using equation 4.2 and 4.3 by all the PoPs where $i=1$ to $P$. Here $P$ is number of PoPs.

131
Figure 4.19: Distributed defense framework

Step 5. The computed entropies in step 4 are sent by all the PoPs to PoP Ps with sum $S_i$ of all the packets received at respective PoP where $i=1$ to $P$.

Step 6. At PoP Ps cumulative source IP $H_S(src\_IP)$, 16-bit traffic cluster $H_S(tc16\_ID)$ and 24-bit traffic cluster $H_S(tc24\_ID)$ entropies are computed using equation 4.9 and 4.10.
Step 7. Source IP $H_s(src_{IF})$, 16-bit traffic cluster $H_s(tc_{16_{ID}})$ and 24-bit traffic cluster $H_s(tc_{24_{ID}})$ entropies computed in step 6 are compared with baseline respective entropies. The detection procedure outlined in figure 4.8 flags normal, flash event, and different types of DDoS attacks.

Step 8. In case DDoS attack is detected in step 7 then characterisation module as per figure 4.14 takes the control and generates attack signatures.

Step 9. A packet having information of all abnormal traffic clusters is encapsulated by PoP Ps and is communicated to all the PoPs which share the multicast group with PoP Ps.

Step 10. All the PoPs detach information of all abnormal traffic clusters from the packet communicated in step 9 and store the same in filter database as attack signatures.

Step 11. Each packet destined to protected web server is allowed to pass only after comparing it with attack signatures stored in filter database.

Thus all attack packets are filtered at the PoPs of the protected ISP domain.

### 4.8 Conclusion

The detection approach proposed in this chapter relies on both source IP address entropy and traffic cluster entropy. The increased dispersion in traffic distribution of source IP addresses i.e. increased source IP address entropy signals occurrence of DDoS attack or FE. But increase in traffic cluster entropy clearly differentiates FE from DDoS attacks due to well known characteristics of FEs highlighted in literature. Moreover, skewness in traffic distribution of source IP addresses i.e. decrease in source IP address entropy clearly provides hints for volume based attacks. Detection flowchart with illustrations of scenarios manifests supremacy of our detection approach. Characterization and filtration modules are also explained by highlighting their logic in flowchart. An ISP based DDoS defense framework is proposed as autonomy, adequate infrastructure, and viability of deployment in incremental fashion is possible in ISP domain. Moreover computational burden is distributed among PoPs of the protected ISP domain so as to make the approach DoS-resistant.