Cooperative Mitigation of DDoS Attacks Using an Optimized Auction Scheme on Cache Servers

Prachi Gulihar[1], B.B. Gupta[2]

[1,2]National Institute of Technology, Kurukshetra, India
[1]prachigulihar2@gmail.com, [2]brij.gupta@gmail.com

Abstract. Distributed Denial of Service (DDoS) attack is one of the most prevalent attacks on the internet today which attacks the availability of the server by resource and bandwidth depletion exhaustion. Many mechanisms exist to fight against DDoS attack, a set of which are the cooperative defense mechanisms which work in a distributed manner and are more robust**.** This work makes use of one of the latest meta-heuristic optimization techniques, Whale Optimization Algorithm (WOA) to find underutilized internet cache servers which are in best position to absorb DDoS flood. These multiple caches will absorb a part of the attack flood thus preventing the victim's network from getting congested. For effective allocation of these cache resources a Continuous Double Auction (CDA) mechanism is applied. It is more flexible and efficient as it allows simultaneous bidding by sellers and buyers. The cache servers are selected through multi-objective WOA in MATLAB and then the auction platform is set-up using Actor Model. In cooperative defense, selection of a pricing strategy which maximizes collateral profit is very important so a round-wise bidding strategy is implemented which promotes long-term participation. For evaluation of the scheme, the workload traces of distributed servers are used to generate three scenarios under different attack load conditions. Depending on the supply-demand of free cache resources, the results show that the proposed algorithm has high detection rate of close optimum solutions. This leads to increased throughput because the attack traffic is not only shared, but is shared in a balanced way.

Keywords: Flooding distributed denial of service attack**,** Cooperative defense, DDoS mitigation, Resource allocation

1 Introduction

Distributed denial of service(DDoS) attack [1] is one of the biggest challenges faced by the internet community today. They are performed by the slave machines which are a part of botnet army and act on the commands of the master machine whose motive is to exhaust network and server resources like bandwidth and storage so that its services become unavailable to the legitimate clients. The largest reported DDoS attack was of volume 400 Gpbs in year 2014 [2]. Since then the DDoS attacks are growing in volume. Their efficiency and implementation techniques have become more

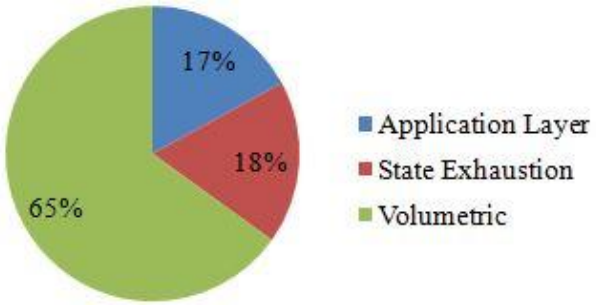

Fig. 1. Types of DDoS attacks

sophisticated day by day making it a big challenge for the security professionals. Figure 1 shows the distribution of various kinds of DDoS attacks the systems are prone to, with 65% of the attacks being the volumetric attacks which are mainly caused by floods of User Datagram Protocol and Internet Control Message Protocol packets. This work focuses on defending this volumetric DDoS attack flood.

The DDoS attack scenario can be divided into attack phase, detection phase and response phase. The difference between a DoS attack and a DDoS attack is that a DDoS attack is launched from different machines whereas the DoS attack involves only single attacking machine which makes it difficult to fight against DDoS attack. When the attacker machines perform in cooperation then for the defense mechanism to be strong it should also be in cooperation. In a cooperative mechanism the defense mechanism is performed in cooperation with other nodes which may lie on the victim-end, source-end or in the network. This overcomes the major drawback of the centralised defense mechanism where there is single point of failure when the central kingpin point of defense itself comes under attack. Although many cooperative techniques have been developed, but they are rarely deployed in the real world because the researchers have long ignored the economic incentive part in the working of cooperatives DDoS mechanisms. Due to lack of incremental payment structures the cooperation between the nodes fails. Sometimes the payment structures are non-existent and in some cases the payment structure is in place but the incentives are not lucrative enough for the nodes to share their resources. Figure 2 shows the timeline depicting the evolution of DDoS exploits and their counter defenses starting from software bugs to traffic flood. The recent defense mechanism involves cooperative traffic control.

Internet comprises of several cache servers which may not be fully utilized. These unused cache capacities can be utilized in cooperative DDoS defense. The traffic flood can be diverted to these multiple servers each handling only a fraction of attack traffic thus preventing congestion from the attack flood. This resource is already existing and will incur meagre costs to the parties involved but management of network resources is one of the most essential issues of Internet. The heuristic techniques of optimization have always been the backbone in solving economic engineering problems and so the main task of the double auction mechanism used in this work is not

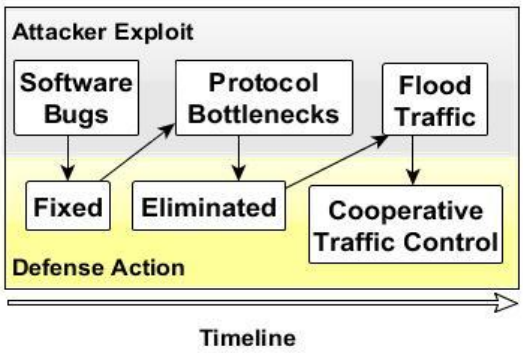

Fig. 2. Evolution of DDoS attacks

only to increase the utility of free cache resources but also promote sustainable individual profits in the long-run. For any cooperative DDoS defense mechanism to be efficient, the resource allocation scheme should be fair any must incentivize the participants to collaborate. The proposed model meets the following design goals [3]:

- *Combination of services:* The marketplace mechanism should allow the users to express complementary requirements like does the victim machine only wants the excess traffic to be diverted and absorbed or it also wants the server to analyze the traffic and report the feedback. Such a combination of services increases the usability of the scheme.
- *Flexibility and predictability:* Depending on the size of the traffic flood at any server, the supply and demand of the free caches in the internet will change dynamically over time. So the buyer will desire an anticipated deal which can be modified and adjusted with changing needs.
- *Economic efficiency:* The provider of the resource and the buyer of the resource both of them desire effective allocation of resources. The policy design should maximize the gains of the participating parties and should minimize the wastage of the resource. So adopting an optimized approach is beneficial.
- *Double-sided competition:* Fair exchange of services should be encouraged between the service providers and the user, the victim machine in this case. The prices should solely depend on the condition of supply and demand and should neither be biased to seller nor to buyer. The market mechanism proposed here is based on double-sided auction model in which the buyers and sellers compete with one another.
- *Functional constraints:* In designing any economic incentive policy, socio-economic objective function needs to be combined with constraints of the network. For optimal results, it is important that the objective function is multi-objective like the basis for the objective functions should depend on multiple attributes like bandwidth, latency, utilization and storage constraints considered while designing the proposed approach.

2 Related Work

DDoS defense mechanisms can be classified on a variety of basis like they can be classified based on their location of deployment and level of cooperation from other nodes as source-end, victim-end and distributed mechanisms. Many methods are already proposed to fight against DDoS attack but no guaranteed defense mechanism exists till now. Numerous solutions have been developed to combat DDoS attack using collaborative schemes in their mechanisms. A few prominent mechanisms suggested till 2018 that catered to formulation of our proposed scheme presented in the following section are discussed below.

Steinberger et al. [4], have proposed a collaborative DDoS defense strategy using flow based event exchange format (FLEX) to exchange event information related to security. They have tried to shift the defense mechanism from victim side to the network of Internet Service Providers (ISP) and their trusted partners. Among several available formats like Incident Object Description Exchange Format, Abuse Reporting Format etc, they have made use of FLEX exchange format in ISPs which makes it possible to deploy this strategy without making any major modifications in the current network infrastructure of network operators.

Rashidi et al. [5], have proposed a collaborative DDoS defense scheme using network function virtualization. Many DDoS defense mechanisms are not hardware compatible with the existing network infrastructure. To overcome this drawback they have made use of emerging technology of network function virtualization by making a domain to domain collaborative network which filters the excessive incoming traffic. For optimal resource allocation they have made use of stakelberg game model. The CoFence framework developed facilitates the resource sharing but it itself may become a target of betrayal, tamper and collusion attacks.

Devi and Yogesh [6], have proposed a hybrid defense mechanism for DDoS attacks at application layer. They have made use of trust information metrics based on information theory. Filtering is done on the basis of the score of the trust value. The rate is further limited based on the user browsing behavior like webpage request order and viewing time elapsed. This two level filtering mechanism gives low false rejection rate and it not only checks the illegitimate traffic but also prevents flooding of legitimate traffic that is flash crowd.

Rodrigues et al. [7], have proposed a cooperative DDOS defense which expands to multiple domains using the signaling process of blockchain. They have overcome a prevalent drawback of existing coordinated DDoS defense mechanism which is signaling of attack information in a distributed environment. The blockchain technology is used for the same which not only signals the information at reduced costs but can also provide financial incentives to cooperate. This happens because of there is no central

point management in the system which makes it suitable for collaborative defense. The major drawback of this scheme is absence of any security mechanism to curb misuse.

Kalkan and Fatih [8], have developed a collaborative filtering based defense mechanism to prevent DDoS attacks. They have proposed a statistical mechanism which filters the traffic based on the current set of attributes like IP address, packet size, destination port number, TTL value , type of protocol and TCP flag . This mechanism called ScoreForCore filters a large volume of the attack packets at the source-end. The score of the packets is calculated based on these attributes and a threshold value is calculated for selective discarding. This cumulative mechanism of scoring is not bound to give accurate results because of dependency on selection of attributes, irrelevant selection may mislead the analysis.

Chunyan et al. [9], have used bloom filters to develop a cooperative DDoS defense scheme which is lightweight. They have deployed two counting bloom filters, the first one differentiates among different network topologies by tracing options field of internet packet. The second filter queries the data collected by filter one to identify the suspicious packets and send alert messages to the victim-end. All the connection topologies possible by the router are analyzed by the first bloom filter and the time complexity of querying is O(1). This method has low processing and memory costs.

2 Proposed Approach

In this section, we describe the different components of the proposed cooperative approach to protect the victim machine from DDoS attack. Figure 3 illustrates the architecture of the proposed defense scheme that can be explained by the following modules.

A. Broker Node: The broker node acts as the first step of defense. All the traffic destined to the victim server cannot directly reach there, it has to pass through the broker node which comprises of the following sub-module.

- *Fetching Module:* The responsibility of this module is to fetch details from the incoming traffic packets. It fetches information like source IP and Destination IP and forwards this information to the analyzing module.
- *Analyzing Module:* This module gets the information of source IP of the packet as input and compares these IP addresses from the IP addresses stored in the log files of the blacklist server. If the address matches, then the packets are not sent to forwarding module. The payload data is also analysed for similarity in packets.
- *Forwarding Module:* This module decides whether to transfer the packets to victim machine or to the optimal resource allocator (ORA) depending upon the congestion levels of the victim machine link. If the threshold level has reached, then the packets are diverted to ORA module instead of victim machine.

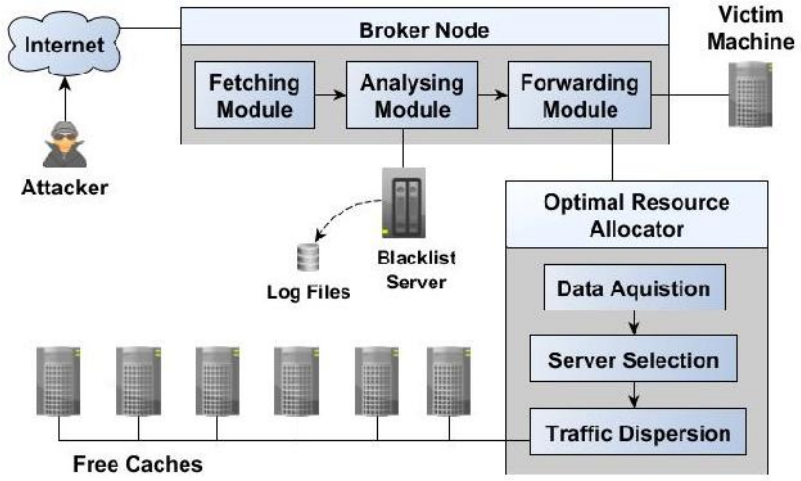

Fig. 3. Architecture of defense scheme

B. Blacklist Server: The blacklist server stores the list of IP addresses which have sent malicious packets in the past. This record is regularly updated and stored in the log files as the traffic arrives at the broker node.

C. Optimal Resource Allocator: When the traffic overflow occurs then the forwarding module sends the incoming packets to the ORA module which is responsible to divert this traffic flood to helping cache servers using the following sub-modules.

- *Data Acquisition:* This is the premier step of the allocation mechanism. In this step the configuration of the cache server like their utilization, latency, capacity and throughput are collected. The initial bid by the victim is set up according to these parameters.
- *Server Selection:* This module is the policy administrator component [10] of the double auction scheme. Its purpose is to select the server in best position to cooperate among the participating cache servers using whale optimization algorithm.
- *Traffic Dispersion:* This module is the policy deployment component which is diverts the traffic to the cooperative cache servers according to the double auction mechanism implemented on the cache servers selected by the administrator component.

D. Free Caches: These are the internet cache servers which are not under full utilization and hence choose to offer their free caches for the cooperative DDoS defense against benefits and economic incentives.

The resource allocation policy used by ORA module can be explained by the following three phases which are executed in a consecutive manner of execution.

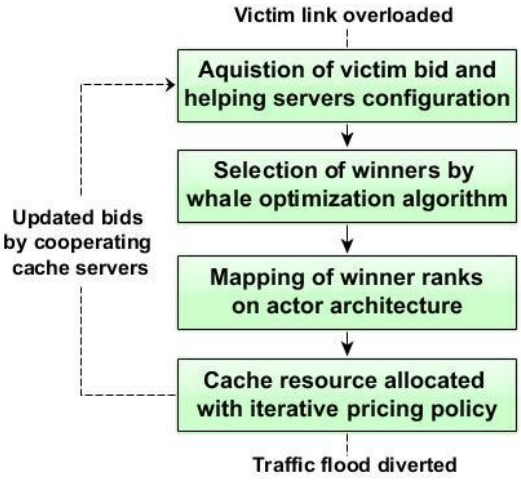

Fig. 4. Flowchart of the proposed algorithm

A. Cache server selection phase: Undertaking minimization fitness function, the operation of optimal selection of cache servers is done between different participating servers. The WOA algorithm which is used for optimal selection is based on hunting strategy of baleen whale. They use the technique of bubble-net feeding in a circular path to hunt the fishes swimming on the surface. This can be mathematically explained by two stages- exploitation and investigation. Firstly, a best candidate is selected which is close optimum and then other operators accordingly modify their location to find a better solution from the available data set. Its working can be explained by the following mathematical equations [11]:

$$D = |C.X'[t] - X[t]| \quad (1)$$

$$X[t+1] = X'[t]-A*D \quad (2)$$

Where t is the current iteration, A and C denote the coefficient vectors, X denotes location vector, X' denotes modified location vector. Further, these vectors A and C are described by the following mathematical equations,

$$A = 2*a*r - a \quad (3)$$

$$C = 2*r \quad (4)$$

Where r ranges from [0,1] and a ranges from [2,0] depicting shrinking of the encircling system in both exploitation and investigation stage.

B. Resource allocation phase: For allocation of free cache memory the decision can be made problem using two different approaches dynamic and greedy. The policy

administrator component stores the values of the costs offered by different servers and the utility function needs to be maximized for maximum benefit. In this policy we have used greedy approach over dynamic because the greedy approach finds the best local solution at every stage leading to a global solution in the end.

C. Iterative pricing phase: The analysis of any policy from a brokering point of view is very essential. Each cooperating cache server has a bid price which is higher than the incurred price of maintaining the cache server. If the server increases its bid price, then although the expected profit will be increased but the winning probability gets decreased and vice-versa. Therefore, an adaptive bidding strategy will be useful.

In designing any incentive mechanism scheme, there exist the problem of incentive cost explosion and there is a risk of servers losing interest in the long run if the incentives received are not at par with Return On Investment (ROI). The dissatisfied servers will not cooperate leading to monopoly in market. A countermeasure of this problem is giving participatory rewards. As shown in the ORA Algorithm, P is the participation credit which is rewarded to the cache server which fails to get a deal. This keeps it motivated to pool the resource. This participation credit is also used by helping server to further lower its bid price, hence increasing its wining probability in next auction round.

Algorithm 1: Broker Module

Input: Incoming traffic X_{in} havingpackets Pk

V[t]: Traffic volume at current instant,

V_{max}: Maximum capacity of channel

Start

Fetch (Pk header, V[t])

If(Source_address[Pk_i]$\in$blacklist_log

&&payload[Pk_i]==payload[Pk_j])

{ Alert(); //malicious behavior

Drop();

Update_log();}

Else Fwd_module(){ **If** (V[t] <V_{max}) //normal flow

{Fwd_server()

{Send[X_{in}] -> server;}}

Else Fwd_ORA(); //overflow

Stop

Algorithm 2: ORA Module

```
Input: Cache servers Cs_i, configuration(u,m,t)
      Where, u= server utilization
            m= free cache, t= throughput
Start   WOA(u, m, t);
      fitness = u + (-m) + (-b); // objective function
            If m_reqd> m
                m = -infinity;
                  Else     m = absolute(m_reqd - m);
            If t_reqd> t
                t = -infinity;
                  Else     t = absolute(t_reqd - t);
Add Cs_i ->winnerlist;                    //optimal servers
Send[winnerlist] ->Auction();
Auction(){Fetch(Rank, winnerlist);
 //utility function at policy administrator component
Utility= (bid_price – incurred_price) * 1/Rank;
Disperse_traffic[X_in] -> Max(Utility[Cs_i])
For all Cs_i
       //iterative pricing at policy deployment component
      If (Cache_NotAllocated)
{P[next_round]=P[previousround]+Incentive[current_round];
            Send(Participation_Credit P)->Cs_i;
            Update_bid()
                  {     New_bid= old_bid – P;
                        Proceed(new_bid);}}
      Else If (Cache_Allocated)
      {     Incentive[current_round]=NULL;
            Proceed(old_bid);}
Stop
```

3 Results and Discussion

We build a prototype of continuous double-auction mechanism using Actor Architecture (AA) [12] and extensive analysis of the double auction mechanism of cache trading is done with key objectives to provide crucial insights on mitigation of the traffic flood. The design of the simulation analysis comprises of Bitbrains dataset [13] of 1250 server machines whose configuration is used for optimal allocation. To select the optimal caches, the MATLAB R2013a version is used in the laptop of Intel COR i3 processor, RAM 4GB, operating system-64 bit. The schedule of workflows is preprocessed and is fed to the whale algorithm and the results are stored in a CSV file which is inputted to the AA using Engine API.

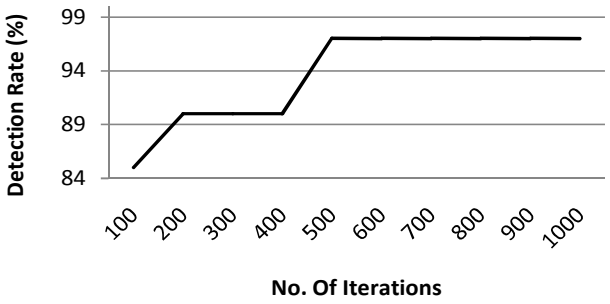

Fig. 5. Detection Rate vs. Number if Iterations

Fig. 6. Throughput vs. Number of Iterations

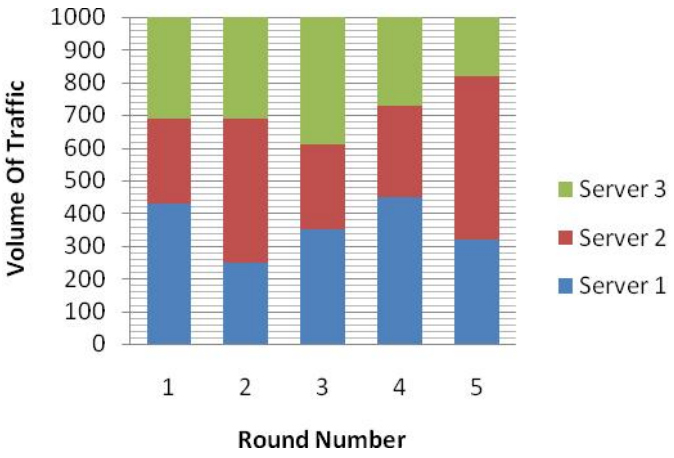

Fig.7. Distribution of attack traffic among helping servers

The above results illustrate a significant convergence in the detection rate of the optimal cache servers. The detection rate of optimal servers in locality is shown in figure 5. The throughput variation of WOA with simulation time of about 3.6 seconds for 1000 iterations is shown in figure 6. Different volumes of cache are traded under different load conditions and an instance of the round wise distribution of attack traffic among different helping cache servers is shown in figure 7. This graph shows that the attack traffic is not only diverted to the cache servers but the diversion is done in a proportional and balanced manner.

4 Conclusion and Future Work

The DDoS attacks are evolving with time and the attackers are creating new ways to exhaust the resources. In this paper we have proposed a cooperative DDoS defense scheme where the victim network redirects the excessive traffic flood to other collaborative internet cache servers with partially filled caches. We have specifically focused on resource allocation mechanism which determines how much cache should the cache servers offer to the victim servers so that free cache resource is distributed fairly, efficiently and with incentives to participate in collaborative defense mechanism. In order to make the allocation of resources optimal we used whale optimization algorithm which finds out the cache servers in best position to help on the basis of attributes like bandwidth, latency, memory and server utilization. To make the collaboration fair, we propose a continuous double auction scheme which allows the both victim server and helping servers to offers bids for the free caches at any moment. The simulation results demonstrate that cooperative DDoS defense effectively mitigates the

traffic and reduces the impact of the attack and the proposed resource allocation mechanism meets the design goal. Long term participation of the helping servers is ensured by offering incentives in the form of participation credit which improves the collateral profit and this mechanism is easily deployable since the cache servers already exist in the internet network.

References

1. Gupta, B. B., Joshi, R. C., &Misra, M. (2009). Defending against distributed denial of service attacks: issues and challenges. Information Security Journal: A Global Perspective, 18(5), 224-247.
2. https://www.calyptix.com/top-threats/ddos-attacks-101-types-targets-motivations/ [Last access on 21/03/2018].
3. Fujiwara, I. (2012). Study on combinatorial auction mechanism for resource allocation in cloud computing environment.
4. Steinberger, J., Kuhnert, B., Sperotto, A., Baier, H., &Pras, A. (2016, April). Collaborative DDoS defense using flow-based security event information. In Network Operations and Management Symposium (NOMS), 2016 IEEE/IFIP (pp. 516-522). IEEE.
5. Rashidi, B., Fung, C., &Bertino, E. (2017). A collaborative ddos defense framework using network function virtualization. IEEE Transactions on Information Forensics and Security, 12(10), 2483-2497.
6. Devi, S. R., &Yogesh, P. (2012). A hybrid approach to counter application layer DDoS attacks. International Journal on Cryptography and Information Security (IJCIS), 2(2).
7. Rodrigues, B., Bocek, T., & Stiller, B. (2017). Enabling a Cooperative, Multi-domain DDoS Defense by a Blockchain Signaling System (BloSS). Semantic Scholar.
8. Kalkan, K., &Alagöz, F. (2016). A distributed filtering mechanism against DDoS attacks: ScoreForCore. Computer Networks, 108, 199-209.
9. Shuai, C., Jiang, J., & Ouyang, X. (2012). A lightweight cooperative detection framework odfDDoS/DoS attacks based on counting bloom filter. Journal of Theoretical & Applied Information Technology, 45(1).
10. Fortier, D., Spradlin, J. C., Sigroha, P., & Fulton, A. (2014). U.S. Patent No. 8,909,751. Washington, DC: U.S. Patent and Trademark Office
11. Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. Advances in Engineering Software, 95, 51-67.
12. Jang, M. W. (2004). The actor architecture manual. Department of Computer Science, University of Illinois at Urbana-Champaign.
13. A. Iosup, H. Li, M. Jan, S. Anoep, C. Dumitrescu, L. Wolters, and D. H. J. Epem (2008). “The grid workloads archive,” FGCS, vol. 24, no. 7, pp. 672–686.