Optimal Sensor Placement for Detection against Distributed Denial of Service Attacks

M.H. Islam[1], K. Nadeem[2] and S. A Khan[3]

[1] Center for Advanced Studies in Engineering, Islamabad, Pakistan *mhasanislam@gmail.com*
[2] Center for Advanced Studies in Engineering, Islamabad, Pakistan *kamran.nadeem@gmail.com*
[3] Center for Advanced Studies in Engineering, Islamabad, Pakistan *shoab@case.edu.pk*

Abstract

Distributed denial of service (DDoS) attacks have become a major threat to organizations and especially to internet and intranet. In DDoS attacks targets are overwhelmed by sending an enormous amount of traffic from a number of attack sites. The major task of any defense system is to detect these attacks accurately and quickly, before it causes an unrecoverable loss. Most of the research in this regard has been focused on the detection techniques without exploiting spatial placement of detection system in a network. The ideal way to completely eliminate the DDoS threat is to run detection mechanism on every node in the network, which is not a practical solution. In this paper, we focus on the optimized placement of detection nodes in a network for distributed detection of DDoS attacks which not only minimize the number of these node required but also reduce the cost, processing overheads and larger delays in identifying an attack. We examine the placement problem of finding a minimum cardinality set of nodes to detect DDoS attacks such that no attack traffic can reach the target without being monitored by these sensors. The placement problem is first formulated as set packing and then as set covering. The solution to both of these formulations is NP hard; therefore, two efficient heuristic algorithms are presented and compared for minimizing the number of detection nodes and finding the optimal placement in a network, thus preventing the impact of distributed attacks. Both algorithms give a near optimal number of detection nodes.

Key words: DDoS, Set packing, Set covering, Security

1. Introduction

Distributed Denial of Service (DDoS) attack is a large scale coordinated attack aimed at overwhelming the victim by sending a vast amount of traffic from multiple sites. As a result the victim exhausts its key resources in processing the attack packets and cannot provide services to its users. During heavy attacks the excessive traffic produced can also make the network heavily congested. DDoS attacks have few characteristics that make them very challenging to detect and defend against: (i) launch from multiple sites; (ii) DDoS attacks do not have common parameters and an expert attacker may change the attack pattern frequently in order to disguise the attack traffic patterns into the legitimate traffic; (iii) attack traffic streams stemming from attack sources are of smaller degree but converge to enormous volume of traffic as they approaches the target.

The only way to completely eliminate the DDoS threat is to make every node secure, which is not a practical solution. There are many challenging tasks involving a variety of algorithmic and engineering design issues in designing an effective and deployable DDoS detection architecture [1], for example, what is the minimum number of nodes required to detect an attack effectively in a network? Generally a large number of detection nodes result into higher costs and communication overheads in reaching to a decision about DDoS attack detection. There are also three goals mentioned in [2] that a practical DDoS defense must meet: (i) accurate attack detection (ii) effective response to reduce flooding (iii) ability to distinguish legitimate traffic from the malicious traffic. These goals can be best met at diverse points in the network. For example to detect the attack accurately a detection node should be deployed in the vicinity of the target [2]. However, it is also often too late to detect the DDoS attack at the victim because it is already overwhelmed by the attack traffic and unable to respond. Ideally, the attack should be detected as close to the sources as possible, saving network resources and reducing congestion. But as we move closer to the source of attack, there are no common characteristics of DDoS streams that can be used to detect the attacks [3]. Moreover, DDoS attacks are launched from multiple sites, so we need to place the detection nodes at various strategic places in the network.

To effectively control the flooding of traffic through the network, deployment of detection nodes in the core is recommended as these nodes can monitor most of the traffic passing through the network [2][4].

Precise traffic identification requires lot of resources because of variability and amount of traffic and can be best met close to the victim because patterns of attack traffic are more significant there. But during the attack a victim experiences huge amount of traffic which limit its abilities to detect the malicious traffic [5][6]. Core routers continuously handle the voluminous and diversified traffic and have limited resource which cannot be dedicated to traffic classification. On the other hand, If the detection nodes are placed near the source, they will face moderate traffic which will require less resource but as we have already mentioned that attack patterns are not significant close to the source and quite a large attack traffic go undetected.

To balance this tradeoff, authors of a cooperative detection approach against DDoS attacks [7] suggested detecting the DDoS attacks in the intermediate network. But as the traffic is not aggregated enough in the intermediate network, a single DDoS detection system cannot detect the attack accurately. This again highlights the requirement of a distributed DDoS detection system with strategically placed detection nodes.

From the above discussion we can conclude that to detect a DDoS attack effectively a distributed DDoS detection system must possess the following characteristics:

- Distributed infrastructure composed of diverse collection of detection nodes
- Detection nodes must be strategically placed
- The number of detection nodes must be minimum
- Collaborative defense, including nodes at all deployment locations
- The sum of partial view of traffic observed by each component can give the overall traffic distribution in the network

We make assumption that the aggregates attack flows towards the victim are larger than the normal flows towards the victim. We are justified in making this assumption because an attacker will always recruit large number of node to launch an attack with very high volume of traffic.

In this paper, we emphasized that optimal placement is a critical factor related to the overall detection system. The focus of this research is to develop methods of placing the detection nodes at optimal locations so as to minimize the overall detection nodes but still ensuring that these nodes can detect most of the attack scenarios. The other advantage of our approach is reducing the computational overhead of security analysis of overall network to a few key nodes.

In order to establish the importance of placement, consider the network shown in Figure 1. Each node in the network can be interpreted as a router or an autonomous system and running the same algorithm. The nodes over which detection will be performed are termed as detection nodes or DN. We assume that all traffic (including both legitimate and malicious) originating from any node and destined for a

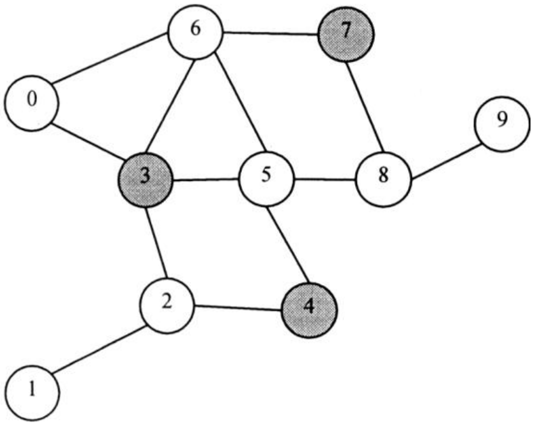

Figure 1. Illustration of DN placement at 3, 4, and 7

node at least *r* hops away (in this illustration we have taken *r =2*) will pass through at least one DN. Routing is performed through the shortest path between the two nodes. In Figure 1, if the DNs are assumed to be placed at nodes 3, 4, and 7 then all the traffic destined to node 1 can be detected except the traffic coming from node 2.

Similarly, malicious traffic coming from nodes 0, 6, and 8, 9 can reach node 5 without passing through any DN. Thus, node 1 is more protected then node 5. But if we place another DN at node 5 then no node in this network can send attack packets to any other nodes that are separated by more than 2-hops. This example illustrates that even a few number of DNs, if optimally placed, can protect a network from attack traffic.

For a detection system to be effective and deployable, the key objectives of placing detection components can be summarized as (i) minimize the total number of DNs; (ii) minimize the total number of nodes that can send the attack packets to any other nodes that are separated by more than the given number of hops without passing through any of the DNs; (iii) find the optimal placement of the DNs that satisfies (i) and (ii).

In this paper, we will focus on two approaches in the class of covering problems, set covering and set packing. The placement problems is first formulated as set packing and then as set covering and both are proven NP hard [8][9]. Therefore, efficient heuristic algorithms are presented to find the optimal placement solution.

2. Related Work

Many research projects and commercial products attempt to tackle the DDoS problem. Only those that provide some form of distributed and hierarchical defense mechanism are reviewed here.

Most of the existing distributed DDoS detection schemes process the data centrally, despite distributed data collection, which limits their scalability and increases the chance of single point failure. To overcome this shortcoming, hierarchical designs were introduced. These systems have layered architecture where attacking events are filtered and preprocessed before they are forwarded to higher levels of the control hierarchy. However, all of these schemes use dedicated nodes and none of these systems talk about optimal number of nodes to be employed in detection system and their optimized placement thus bringing restrictions and communication overhead to their functionality. Recent studies have also shown that there is no single deployment point that can successfully fulfill the requirements of an effective DDoS detection system. An overview of the work related to us is appended below.

Distributed defense against DDoS attacks [2] discussed the importance of placing detection sensors at diverse points in the network, namely close to the victim for accurate detection, in core of the network for effective flood control and close to the source for precise traffic identification. All these nodes form an overlay network and interact with each other when under attack. It does not discuss any algorithm or optimization regarding the selection of these nodes.

Attacking DDoS at source [5] proposed a DDoS defense system deployed at attack source-end networks that autonomously detects and stops attacks originating from these networks. Attacks are detected by carrying out periodic comparisons of two way traffic flows between the network and rest of the Internet with normal flow models.

Cooperative mechanism against DDoS attacks [7] is based on the fact that DDoS streams do not have common characteristics; therefore, currently available intrusion detection systems (IDS) cannot detect them accurately. They propose a distributed approach to detect distributed denial of service attacks by coordinating across the Internet. Unlike traditional IDS their detection is carried out at the intermediate network. The nodes involve in the detection process, form an overlay network and communicate through gossip protocol.

The work presented by Seok, Young, and Sehun [10] is closely related to our proposal. The detection system placement problem is formulated as set packing problem and is able to localize the attack traffic within r hops. The heuristic proposed is complex as compared to our work. Moreover, we have been able to identify few deficiencies in their work. This is discussed in detail in the section of performance comparison. The topology and some mathematical notations used to illustrate the importance of optimal placement are similar to the one found here for establishing a direct relationship and a more meaningful comparison

A gateway-based defense system for DDoS attacks in high-speed networks [11] discussed a gateway-based approach. A set of gateways are deployed at different locations in the network to collaboratively perform the desired countermeasure functions such as attack detection and traffic access control.

Optimal allocation of filters against DDoS attacks [12] optimally allocates filters available in a single router to attack sources, or entire domains of attack sources, so as to maximize the amount of good traffic preserved, under a constraint on the number of filters. They have formulated two filtering problems: the single-tier and the two-tier filtering, depending on the granularity of the packet filtering. Filter allocation in single-tier is formulated as knapsack problem and in two-tier they used dynamic programming. In single-tier filtering is performed on the entire gateway whereas in two-tier filtering is carried out not only on the entire gateway but also on the individual attackers.

A packet filter placement problem with application to defense against spoofed denial of service attacks presented in [13] relates the filter placement problem to the vertex cover problem. Their approach does not explicitly covers the majority of DDoS flooding attacks but only deals with spoofed denial of service attacks. Routing tables are extensively used by the two types of filters, maximal and semi-maximal, defined in their work. This scheme can have degraded performance if routing tables are well populated specially in the case of large scale networks.

In divide-and-conquer strategy for thwarting distributed denial-of-service attacks [14] detection is performed near the victim host and packet filtering is executed close to the attack sources. The process isolates one attacker and throttles it. The parallel version of this technique is capable of throttling traffic coming from a large number of attackers simultaneously. This technique works along the flow of traffic and considers only those routers that come in it. Processing overhead and communication among the detecting nodes and filtering nodes can be increased considerably if the number of attackers is large and lunching attacks from various multiple sites.

Collaborative detection of DDoS attacks over multiple network domains [15] suggested to detect abrupt traffic changes across multiple network domains. Each domain has a change aggregation tree (CAT) servers to aggregate the flooding alerts reported by the routers. CAT domain servers collaborate among themselves to make the final decision.

3. Problem Formulation through Set Packing

Set packing problem comes up in partitioning applications, where we need to partition the elements comprising the network under the strong constraints on what is an allowable partition. The key feature of set packing problem is that no nodes are permitted to be the member of more than one set. In keeping up the correspondence with the computer networks, elements can be considered as nodes (routers or autonomous systems, and running the same algorithm) and partition constraint as the distance between the two nodes in terms of hops. Our objective is to partition the network into large subsets of nodes such that each edge is adjacent to at most one of the selected nodes and all nodes not adhering to the partition constraint but at least adjacent to one of the node in the set are declared as detection nodes (DN).

We model the network as finite connected undirected graph, $G\ (V, E)$, where V is the total number of nodes comprising the network and E is the total number of edges making a network. Each node in V can be interpreted as a router or autonomous system and has the capability of performing the function of DN (detection node) when elected to do so. Moreover, same algorithm is running on each node. An edge in E is defined for a node pair (i, j) if and only if i and j are directly connected to each other and can exchange messages without going through any other node.

Let r be the tolerance level against attack. That is, any node less than r hop away is permitted to attack another node because the impact of attack of these nodes can be small enough to be ignored when r is small. To calculate the detecting effect, $R_i\ (r)$ is defined as the risk level of a node i against attacks, as the number of nodes that are more than r hops apart from node i and can send attack packets to node i without passing through any of the DN. If $R_i\ (r) = 0$, it means there is no node beyond r hops which can send attack packet to node i, and every attack can be localized within a set of nodes, less than r hops from node i. Moreover, as all traffic targeted to node i must pass through at least one DN, so all attack packets targeted to node i would be detected provided if $R_i\ (1) = 0$.

We use T to denote a subset of nodes (DNs) where detection is performed such that $T \subseteq V$. Our goal is to select a small subset of the nodes T that can act as DNs for the network such that every other node is at least r hops away from at least one member of the subset T. Such a set is known in graph theoretic terminology, as a dominating set (DS). In this work, our focus is also on minimizing the cardinality of the dominating set i.e., on picking as few DNs as possible. Such dominating sets are called minimum dominating sets (MDS). Unfortunately, finding the domination number of the graph is a difficult problem and the decision version of the problem is NP hard [9]. Thus, our goal changes to finding efficient heuristic to the MDS problem.

We also defined $\tau = |T| \ / \ |V|$ as the coverage ratio, between the number of DNs required to provide cover to the total number of nodes comprising the network. Obviously, we would like to keep the value of τ to as minimum as possible for a given value of r hops.

. Given a finite set $V = \{1, \ldots, m\}$ be the finite node set and $V_j = \{V_1, V_2, \ldots, V_n\}$ is given as a collection of subsets of V *(i.e.,* $V_j \subseteq V, j=1,\ldots,n)$ and a set packing P is formed with respect to V if $V_j \cap V_k = \Phi$ for all j, k and $j \neq k$, and $M(P)$ represents the number of nodes comprising a set packing and $N(P)$represents the totals number of set packing formed. The maximum value of $d\ (i, j)$ for all nodes $i, j \in V_k$ in a packing should be less than r. Maximizing the number of nodes in each set packing and also maximizing the number of set packing is equivalent to minimize the number of DNs. We have also observed that a smaller value of r in V yields a larger T. A larger value of r will reduce the T but put the node at a greater risk, because a node can receive attack traffic without being detected by any of the DN. In terms of the model, our objective is to find a set of nodes that are at $(r\text{-}1)$ hops away from each other and such that every other node not in the set is r hop away from at least one node in the set.

Mathematically, we can formulate the problem as follows:

minimum $|T|$ (1)

such that $R_i\ (r) = 0$, *for all* $\forall i \ \in V$

maximum $\sum_{i \in V} \quad M_i(P)$ (2)

for all $\forall i \ \in V$ such that $d(i,\ k) \leq r\text{-}1$ *for* $\forall i \ \in V,\ k \in V_j,\ i \neq k, \forall j \ \in N$

maximum $\sum_{j \in N} \quad N_i(P)$ (3)

for all $\forall i \ \in V$ such that $V_j \cap V_k = \ \Phi$ for all j, k and $j \neq k$

$d(i, k) \geq 2r$ (4)

for all $\forall i \ \in V_i$, $\forall j \ \in \ V_j$, i≠j, *and* $V_j \cap V_k = \Phi$

Equation (1) is our major requirement. Constraints (2) and (3) are general set packing constraints; means that there should be maximum number of nodes in a set pack such that the distance between any two nodes in a set pack should always be less than $r\text{-}1$ and there should be maximum possible set packs. Equation (4) tells us that the distance between two nodes belonging to two different set packs should always be more than 2-hops.

This problem is basically a set packing problem, which is NP-hard [8]; hence, we propose a heuristic algorithm. This algorithm partitions connected nodes in such a way that the maximum value of the distance in the partition is less than r. Then, all nodes connected directly to the partition become DN nodes. This process repeats until all nodes either become member of a partition or become DN nodes.

<Heuristic>

Let V\` and V\`\` are the temporary sets for holding intermediate results.

Initially DN = ∅, V\` = ∅, V\`\`= ∅

Step 1: Select a node n with minimum number of links such that n ∈ V

Step 1.1: V\` = V\` ∪ {n}

```
Step 2: Select a node n` with lowest hop count from n and less than r hops away from n such that $n` \in V; n` \notin V`; n` \notin V``$

Step 2.1: If there is no such node go to Step 3

Step 2.2: $\forall x \in V`$, check whether n` is less than r hops away from x

Step 2.2.1: If yes, $V` = V` \cup \{n'\}$; else $V`` = V`` \cup \{n'\}$

Go to Step 2

Step 3: All nodes directly connected to nodes in V` and not members of V` are added to DN i.e. ($DN = DN \cup \{x\}; \forall x: x \in V, x \notin V`$ and $E(x, n)\ \exists n \in V`$

Step 3.1: $V = V \setminus V`$

Step 3.2: $V` = \emptyset; V`` = \emptyset$

Step 3.3: If $V = \emptyset$; Go to Step 1 else terminate.
```

4. Results

In order to calculate the performance of the proposed algorithm the simulation is run on three different topologies (Figure 2, 3 and 4). Topologies given in Figure 2 and 4 are the same as mentioned in [9]. In order to further elaborate the performance, we have also considered a connection-based scheme (CPS) that selects the nodes as DNs from a network based on the number of connections repeatedly until the target *r* is reached. In case where more than one node has the same number of connections, it chooses the node randomly.

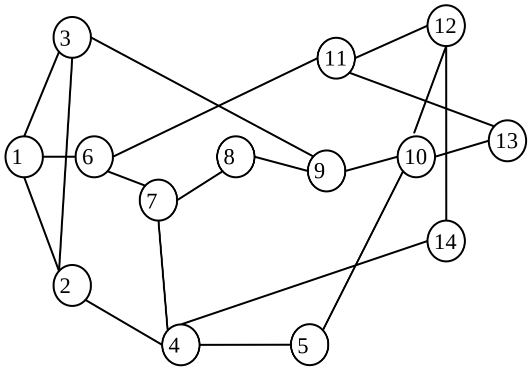

Figure 2. 14 – node network

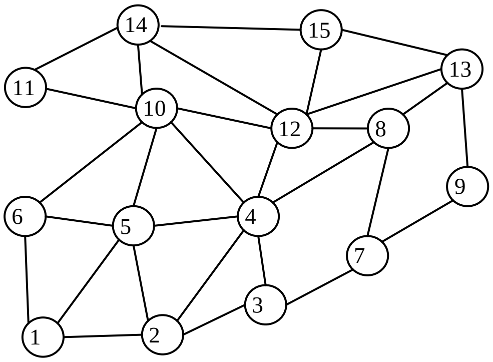

Figure 3. 15 – node network

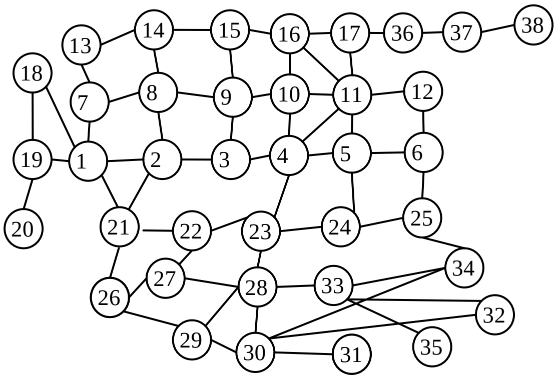

Figure 4. 38 – node network

In our simulations, routing policies are based on multi-path routing and we select all possible loop-free shortest paths between the two nodes when routing is performed. Figure 5, 6, and 7 shows the average coverage ratio of the three different networks as r increases from 1 to 8. In all the three cases performance of CPS is poor. This proves that DN placement is an important issue and a node cannot be selected as DN merely based on connectivity. The proposed scheme performs better than the two (CPS and [9]) and gives lower coverage ratio. This means that our scheme selects fewer nodes as DNs than CPS and the one proposed in [9]. However, it is noted that selecting a large value of *r*, decreases the average coverage but make the network more vulnerable to DDoS attack because it will leave more paths open to the attacker. On the other hand a lower value of *r* gives more coverage ratio (more DNs) but at higher cost processing and communication overhead. For example at *r = 1* majority of nodes in the network will be selected as DN, but that is practically highly inefficient and fails our purpose. Therefore, value of *r* must be chosen carefully keeping in view the current system resources, network configuration and impact of attacks.

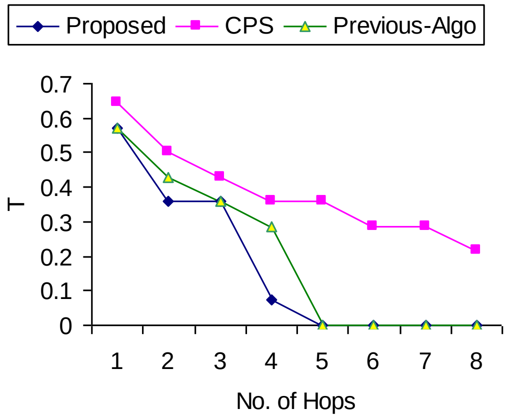

Figure 5. Coverage ratio of 14-node network

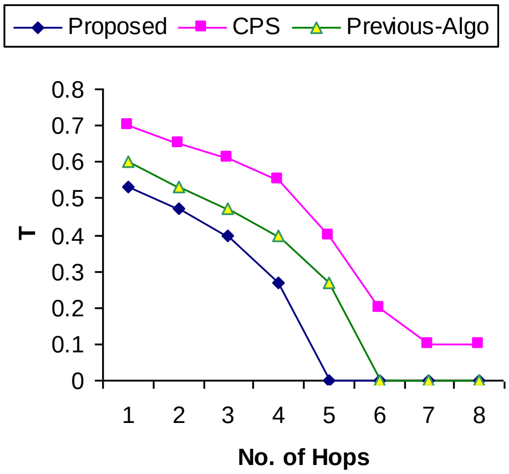

Figure. 6. Coverage ratio of 15-node network

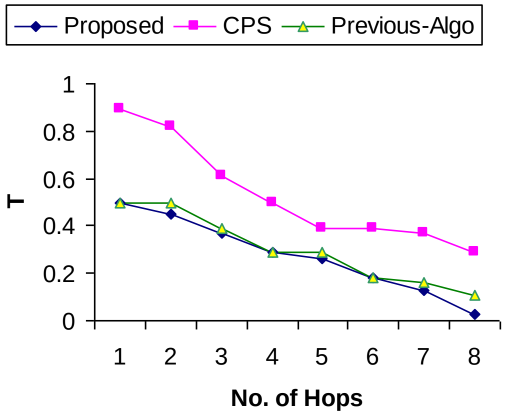

Figure 7. Coverage ratio of 38-node network

We have also observed a special case where more than one node is connected to the node under consideration and both have the same number of edges. The algorithm [10] does not produce accurate results in this case. Consider the Figure 8, for r = 2 and starting from node 0 the algorithm [10] will pick either node 1 or node 2 and accordingly selects node (2, 3) or (1, 3) as DNs and then in the second iteration based on the selection of nodes (4, 5) or (4, 7), or (6, 5), or (6, 7) it will select nodes (6, 7), or (5, 6), or (4, 7), or (4, 5) as DNs. Whereas our proposed algorithm picks (0, 1, 2) and form the group because all these nodes are within *(r-1)* hops with each other. Therefore, in the first iteration node 3 will be marked as DN. Similarly in the second iteration nodes (4, 5, 7) or (5, 6, 7) will form the group and either 4 or 6 will be selected as DN depending upon the group chosen. So the convergence ratio at r = 2 for the algorithm proposed in [10] is 0.5 whereas our algorithm gives 0.25.

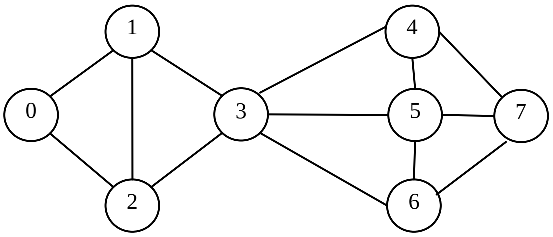

Figure. 8 7-node network

5. Problem Formulation through Set Covering

The set covering problem is a fundamental problem in the class of covering problems. Given a finite set X and a family $F = S_1, S_2, \ldots, S_n$ of subsets of X, (i.e., $S_j \subseteq X, j = 1, \ldots, n)$ the set covering problem is to find a minimum cardinality $J \subseteq \{1, \ldots, n\}$ such that $\bigcup_{j \in J} S_j = X$. The elements of X are called points. Given a $J \subseteq \{1, \ldots, n\}$ a point is said to be covered if it belongs to $\bigcup_{j \in J} S_j$.

In keeping up the correspondence with the computer networks, members of a set can be considered as nodes (routers or autonomous systems, and running the same algorithm) and diameter of a subset formed as the distance between the two nodes in terms of hops and a node is said to be covered if it belongs to a subset. The key feature of set covering problem is that a node can be a member of more than one sub set as oppose to set packing where a node can belong to only one sub set. Our objective is to partition the network into subsets of nodes in a way that each subset can accommodate maximum number of nodes (covering maximum nodes) and all node(s) having the highest connectivity in a partition are declared as detection nodes (DN).

As mentioned before, a network can be defined as a finite connected undirected graph, *G (V, E)* with *V* nodes and *E* edges. Every node runs the same protocol. An edge is defined for a node pair *(i, j)* if and only if *i* and *j* can exchange messages without going through any other node. $V = \{1 \ldots m\}$ be the finite node set and $V_j \subseteq V$, where $j = 1, \ldots, n$ defines a subset of *V*. The set covering problem is to find a minimum cardinality $J \subseteq \{1 \ldots n\}$ such that $\bigcup_{j \in J} V_j = V$. Given a $J \subseteq \{1 \ldots n\}$, a node is said to be covered if it belongs to V_j, $\forall j \in J$.

Before going further, we would like to mention a very significant difference between the formulation and application of set packing and set covering towards the effective placement. In set packing, we ensure that there is a DN after every *r* distance and attack traffic cannot reach to its target without passing through one of the DN (assuming $r = 1$), whereas, in set covering, though, sets are formed based on the value of *r* and then within this set, highest connectivity node is picked (assuming it is connected to all members of it set) as DN but there is no guarantee that attack traffic originating from any of the member of the set will pass through the DN (assuming $r = 1$). In fact, in both the cases increasing the value of *r* will leave a portion of the traffic undetected (few links will remain unsupervised). This leads us to a conclusion that set covering is more appropriate to apply for wireless and ad hoc networks, where, even if the traffic is not passing through the DN but still it can overhear the traffic, provided the nodes involve in exchanging the traffic are in the transmission range of DN. Set packing is a difficult choice for wireless and ad hoc networks because of their dynamic topology, as, it will become very difficult to maintain the constraints of distance and placing a DN after every fixed number of hops on a given path.

Tolerance level *r*, risk level $R_i\ (r)$, number of DNs selected as *T*, coverage ratio τ are same as defined before. Let the neighbors of a node *i* is denoted by *X(i)* and consists of all those nodes which are in the direct communication range of node i $(r = 1)$. Let $X`(i)$ be the closed neighborhood of *i* such that $X`(i) = \{i\}\ U\ X(i)$. Also, for any set $V_j \subseteq V$, let $X\ (V_j) = \bigcup_{j \in S} X`(j)$

A node is said to be covered, if either this node or one of its neighbors become a member of a subset V_j. At the start of the algorithm, no nodes are covered and for any k^{th} iteration, let $X_k^u(i)$ define the number of uncovered neighbors of node *i*, including *i* itself if it is uncovered. Now for each iteration select a node i_k such that it is the most highly connected node in the current iteration and it can exchange messages with all its directly connected neighbors (not necessarily one hop but still i_k can monitor their ongoing traffic). This node (i_k) is the candidate for every one of its uncovered neighbors. We can define the elected node as:-

$$X_k^u\ (i_k) = \text{maximum}_{\ j \in X`(i)}\ X_k^u\ (j)$$

And

$$i = j_k\ , \qquad \forall j \in X_k^u\ (i) \qquad (1)$$

Now according to Greedy [16], put all elected nodes (DNs) in the dominating set in each iteration, and end when all the nodes are covered.

Mathematically, we can formulate it as

$$\text{Minimum} \qquad \sum_{i \in V}^{j \in N} V_{ij} \qquad (2)$$

$$d\ (i, k) \leq r \qquad \textit{for } \forall i,\ \forall k \in V_j \quad (3)$$

$$\text{Minimum } |T| \qquad (4)$$

Such that $R_i\ (r) = 0$ for all $\forall\ i \in V_j$

Equation (2) is the general set covering problem. Equation (3) shows that the distance among the nodes belonging to a subset must always be less than or equal to r. Equation (4) shows that number of DNs selected should be minimum and that only those nodes are qualified to become a DN which have maximum connectivity among their neighbors and satisfy equation (1).

This problem is basically set covering problem which is NP hard [9]. Hence, we have to rely on heuristic solution methods. Our heuristic algorithm partition connected nodes such that nodes forming a set are immediate neighbors (direct communication or *r = 1*), then within each subset a node satisfying (1) will be selected as a DN node. This process repeats until all nodes either become member of a subset or become DN nodes.

Lets U - is the set of nodes comprising a network and initially each node is uncovered

```
<Heuristic >

DN = $\emptyset$.

Step 1: $\forall j \in U$, make subsets $V_j$ of the node set U in such a way that: $V_j = \{j\}$, $V_j = V_j \cup \{x: \exists x \in U$ and x is r hops away from $j\}$.

Set i = 1 to r

    Step 1.1: $V_j = V_j \cup \{x: \exists x \in U$ and x is i hops away from $j\}$

Step 2: Choose a node i such that it has maximum uncovered neighbors including itself (the largest subset available) such that $|V_j \cap U|$ is maximum among all $V_j$.

Step 3: $U = U \setminus V_j$

Step 4: $DN = DN \cup \{i\}$

If $U = \emptyset$, terminate. Else go to Step 2.
```

A. *Results*

In order to calculate the performance of the proposed algorithm the simulation of the proposed scheme is run on the same topologies given in Figure 2, 3, and 4, and a connection-based scheme (CPS) is also considered here for performance comparison as in the case of set packing.

In these simulations routing policies that allow multipath routing are considered and we select all possible loop-free shortest paths whenever the routing is performed between the two nodes. Figure 9, 10, 11 shows the average coverage ratio for all nodes as *r* increase from 1 to 8 and an increase in *r* means coverage area of a node is also increase. From these figures it is obvious that performance of CPS is limited and it gives high coverage ratio (larger value of *T*) as compare to our proposed scheme which gives low coverage ratio (low value of *T*). Thus, our scheme requires fewer numbers of DNs as compare to CPS for the same value of *r*.

Another point to observe is that as the values of *r* increases coverage ratio also decrease, it means network becoming more vulnerable to attack and some of the attack traffic can go un-detected. As the value of *r* approaches to 1, coverage ratio increases, network is less vulnerable and attack detection becomes more accurate (DNs are monitoring maximum traffic). However, perfect attack detection (*r = 1*) is difficult to attain and value of *r* should be chosen based on the current system resources and impact of attack.

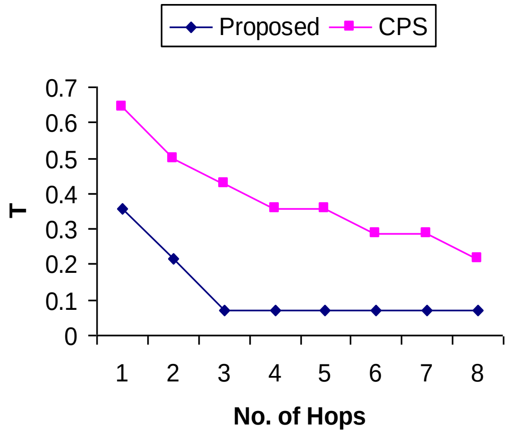

Figure 9. Coverage ratio of 14-node network

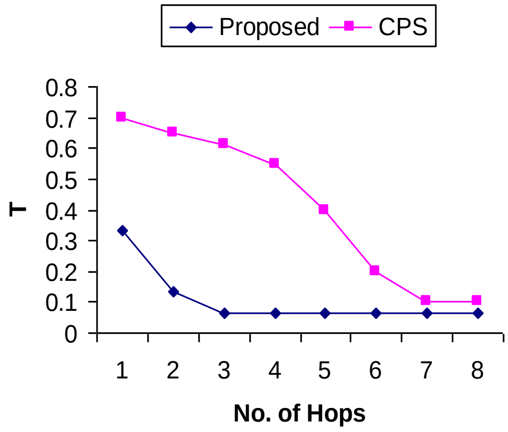

Figure 10. Coverage ratio of 15-node network

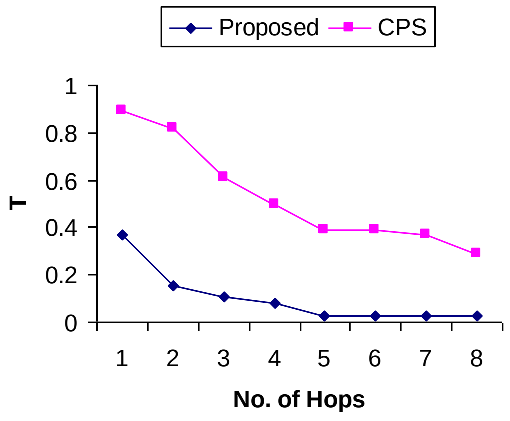

Figure 11. Coverage ratio of 38-node network

1. Performance Analysis

A. Cardinality Properties of the proposed heuristics

Cardinality of a set is defined as the number of members in a set. As we have discussed previously that our objective is to keep the cardinality of the DN set at minimum as possible. Chvatal [16] and others [19], [20] have also analyzed the greedy algorithms for calculating set covers. In their work they compared the cardinality of the sets returned by the algorithm to that of the smallest cover in the worst case. Since any dominating set problem can be formulated as a set covering and as a set packing problem therefore the results of the work carried out by Chvatal [16] can be directly applied here:

$$|D_{sg}| / D_o \leq \sum_{i=1}^{\delta+1} 1/i \qquad (1)$$

Where δ is the maximum degree of a node in the graph, D_{sg} is the size of the dominating set returned by the algorithm and D_o is the cardinality of the minimum dominating set. It is also shown in [18][19] [20] that for an undirected network of N nodes and M links, following upper bound on D_o applies to $|D_{sg}|$ as well (D_{sg} = D_o) and provably close to the minimum cardinality set:

$$D_o \leq N + 1 - \sqrt{(2M+1)} \qquad (2)$$

Table 1 – 3 show the number of nodes that belong to the detection node set. This result describes the performance of the algorithm in terms of dimensions of the graph or the number of DNs selected, and should be viewed as complementary performance parameter to the coverage ratio calculated by our proposed algorithms. From the results mentioned in the tables below, it can be seen that cardinality of a minimum dominating set is always within the range defines by (2) above.

Table 1. DN nodes in a 14 node network

	Number of Hops (r)							
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Set Packing	8	5	5	1	0	0	0	0
Set Covering	5	3	1	1	1	1	1	1

Table 2. DN nodes in a 15 node network

	Number of Hops (r)							
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Set Packing	8	7	6	4	0	0	0	0
Set Covering	5	2	1	1	1	1	1	1

Table 3. DN nodes in a 38 node network

	Number of Hops (r)							
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Set Packing	19	17	14	11	10	7	5	1
Set Covering	14	6	4	3	1	1	1	1

Another important point is that cardinality of set covering is always less than set packing for a given value of *r*, because of the reason that in set covering a node can be a member of two sets at the same time but in set packing a node must be a member of one unique set of nodes. Considering table 1, the set packing algorithm gives more number of DNs as compared to set covering as long as the value of *r* is less than 4. At *r = 4*, both algorithms found only one node and at *r* greater than 4, set packing did not find any DN(s). This is because of the fact that in the 14 node network every node is within 4 hops of each other. As the maximum number of hops in the network is 4, set packing failed to find any detection nodes because the partition formed contained all the nodes in the network and thus the DN set remained empty. The set covering result of the same network, however, nominated at least 1 node no matter what the value of *r* was. The reason is that the node with the maximum number of links in the network provides coverage to the whole network when the value of *r* reaches 3. Same analogy can be applied to table 2 and 3 also.

A. Performance Analysis of Proposed Heuristics

The coverage ratio graphs of both the set packing and set covering algorithms are compared, leading to some interesting observations.

As can be seen from Figure 12, 13 and 14, set covering provides coverage ratio τ which is considerably less than the set packing for the same value of *r*. In other words numbers of DNs returned by set covering are less than the DNs returned by set packing. The reason is the nature of set forms in the two approaches. Set packing makes mutually exclusive sets based on the hop distance *r* such that each DN selected is a member of only one set, whereas in set covering DN selected can be a member of more than one set.

Running simulations on the example network of Figure 1, we find that the nodes nominated as DNs in set packing are 3, 4, 5 and 7 and in set covering 3, 5 (keeping r = 2). Further in the case of set packing it is ensured that no node can send data to a node more than *r* hops away without first encountering a DN on its path to the destination whereas, in the case of set covering this restriction no longer applies. In fact in set packing nodes can send data to other nodes even if they are 3 or 4 hops away without passing through any DN. This distinguishing behavior between the two approaches is led to another conclusion that set packing is more applicable to wired networks, where we want to minimize the number of nodes that could send the malicious traffic to any other nodes that are separated by more than the given number of hops (r) without passing through the DNs. Set covering can provide optimal placement for wireless and ad hoc networks where number of separation *(r)* can be considered equivalent to the coverage area provided by the transmission range of the antenna in the node. The nodes selected as DN can overhear all the traffic taking place in its coverage area.

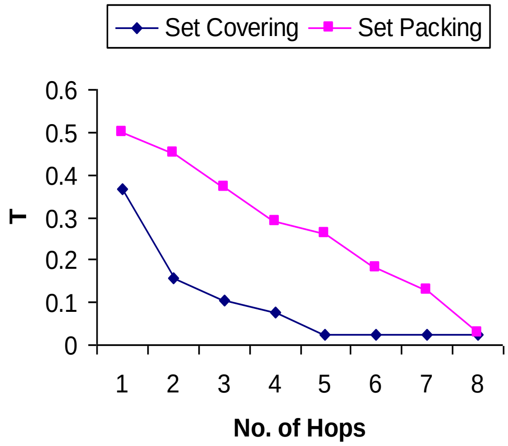

Figure 12. Coverage ratio of 14-node network

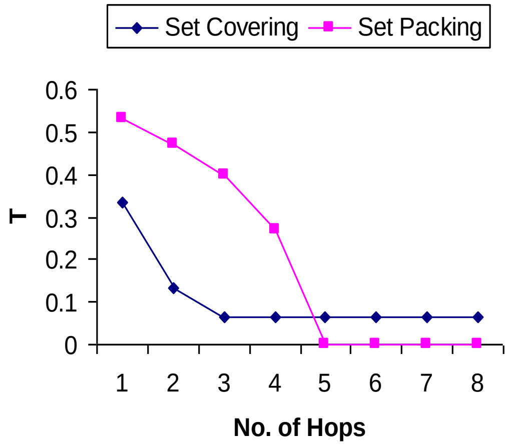

Figure 13. Coverage ratio of 15-node network

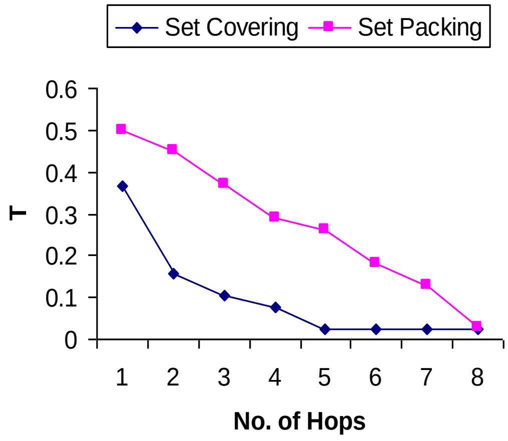

Figure 14. Coverage ratio of 38-node network

Conclusions

It is difficult to achieve a perfect detection against DDoS attacks. The only way to achieve this is to make every node participate in the detection process which is not a feasible solution. This emphasize that there has to be some way to reduce the number of detection nodes but still able to provide the effective protection against DDoS attacks.

Our main contributions are highlighting the importance of placing detection nodes at critical points in a network and reducing the number of nodes participating in the detection process and still localizing the impact of attack with in *r* hops or within the direct transmission range in case of wireless network. Addition advantages can be reduced cost and faster convergence in identifying an attack.

Our main emphasis is this research has been on the architecture of the detection system rather than on the detection technique or collaboration among the nodes. But once the nodes are selected then any existing mechanisms can be employed.

These heuristics are centralized in nature i.e., complete topology of the network is known. But there can be scenarios where a node has only the partial view of the topology (wireless and ad hoc network). For those cases, we aim to develop distributed heuristic where a node do not know the size of the network, and start out with limited topological information of the network. Our goal will be to select a small subset of nodes, without having the complete network information that can act as detection nodes for the ad hoc network.

References

[1] Wan, K.K.K.; Chang, R.K.C., "Engineering of a global defense infrastructure for DDoS attacks," Networks, 2002. ICON 2002. 10th IEEE International Conference on , vol., no., pp. 419-427,2002

[2] Mirkovic, J., Robinson, M., Reiher, P. and Oikonomou, G. “Distributed Defense against DDoS Attacks”, University of Delaware CIS Department Technical Report CIS-TR-2005-2.

[3] CHANG, R. K. C. “Defending against flooding-based, distributed denial-of-service attacks:” A tutorial, *IEEE* Communications Magazine 40(10): 42–51

[4] Aditya A., Ashwin B., Mike Reiter and Srinivasan Seshan, “Detecting DDoS Attacks on ISP Networks”, ACM SIGMOD/PODS Workshop on Management and Processing of Data Streams (MPDS), FCRC 2003, San Diego, CA.

[5] Mirkovic, G. Prier, and P. L. Reiher, “Attacking DDoS at the source,” in Proceedings of the IEEE International Conference on Network Protocols (ICNP '02), pp. 312–321, Paris, France, November 2002

[6] Mirkovic, J. and Reiher, P., D-WARD: A Source-End Defense Against Flooding Denial-of-Service Attacks, IEEE Transactions on Dependable and Secure Computing, Vol. 2, No. 3, July-September 2005, pp. 216-232

[7] Guangsen Z., Manish, P: Cooperative Defense against DDoS Attacks. Journal of Research and Practice in Information Technology 38(1): (2006)

[8] Garey, M.R., Johson, D.S., “Computers and Intractability: a Guide to the Theory of NP completeness”, W.H. Freeman and Company, San Francisco (1979)

[9] Gallager, R. G. “Distributed Minimum Hop Algorithms”, Tech. Rep. P1175, MIT Laboratory for Information and Decision Systems, 1982

[10] Seok B.J., Young W.C., Sehum K.“An Effective Placement of Detection Systems for Distributed Attack Detection in Large Scale Network” WISA 2004, LNCS 3325, pp. s204-210, 2004

[11] Dong, X., Riccardo, B., and Wei, Z.o, A Gateway-Based Defense System for Distributed DoS Attacks in High Speed Networks, in Proc. of IEEE Systems, Man, and Cybernetics Information Assurance Workshop (IAW), June 2001, pp. 212-219

[12] El Defrawy, K.; Markopoulou, A.; Argyraki, K., "Optimal Allocation of Filters against DDoS Attacks," Information Theory and Applications Workshop, 2007 , vol., no., pp.140-149, Jan. 29 2007-Feb. 2 2007

[13] Benjamin, A., Cole S.J., Kihong, P.; “A Packet Filter Placement Problem with Application to Defense Against Spoofed Denial of Service Attacks”, European Journal of Operation Research, Volume 176, Issue 2, 16 January 2007, Pages 1283-1292

[14] Chvatal, V.; “A Greedy Heuristic for the Set-Covering Problem”, Math. of OR 4:3 (1979), pp. 233-235

[15] Hochbaun, D. S.; “Approximate Algorithms for the Set Covering and Vertex Covering Problems”, SIAM Journal on Computing,11 (1982), pp. 555-556

[16] Johnson, D. S.; “Approximate Algorithms for Combinatorial Problems”, Journal of Computer System Science, 9 (1974), pp. 256-278

[17] Parekh, A. K.; “Analysis of Greedy Heuristic for Finding Small Dominating Sets in Graphs”, Information processing letters, 39 (1991), pp. 237-240

[18] Vizing, V.G.; “A Bound on the External Stability Number of a Graph”, Doklady A.N., 1964 (1965), pp.729-731

[19] Ruiliang, C., Jung-Min, P., Randolph, M.; "A Divide-and-Conquer Strategy for Thwarting Distributed Denial-of-Service Attacks," IEEE Transactions on Parallel and Distributed Systems, vol. 18, no. 5, pp. 577-588, May, 2007

[20] Yu, C.; Kai, H.; Wei-Shinn K., “Collaborative Detection of DDoS Attacks over Multiple Network Domains” IEEE Transactions on Parallel and Distributed Systems, Volume 18, Issue 12, Page(s):1649 – 1662, Dec. 2007